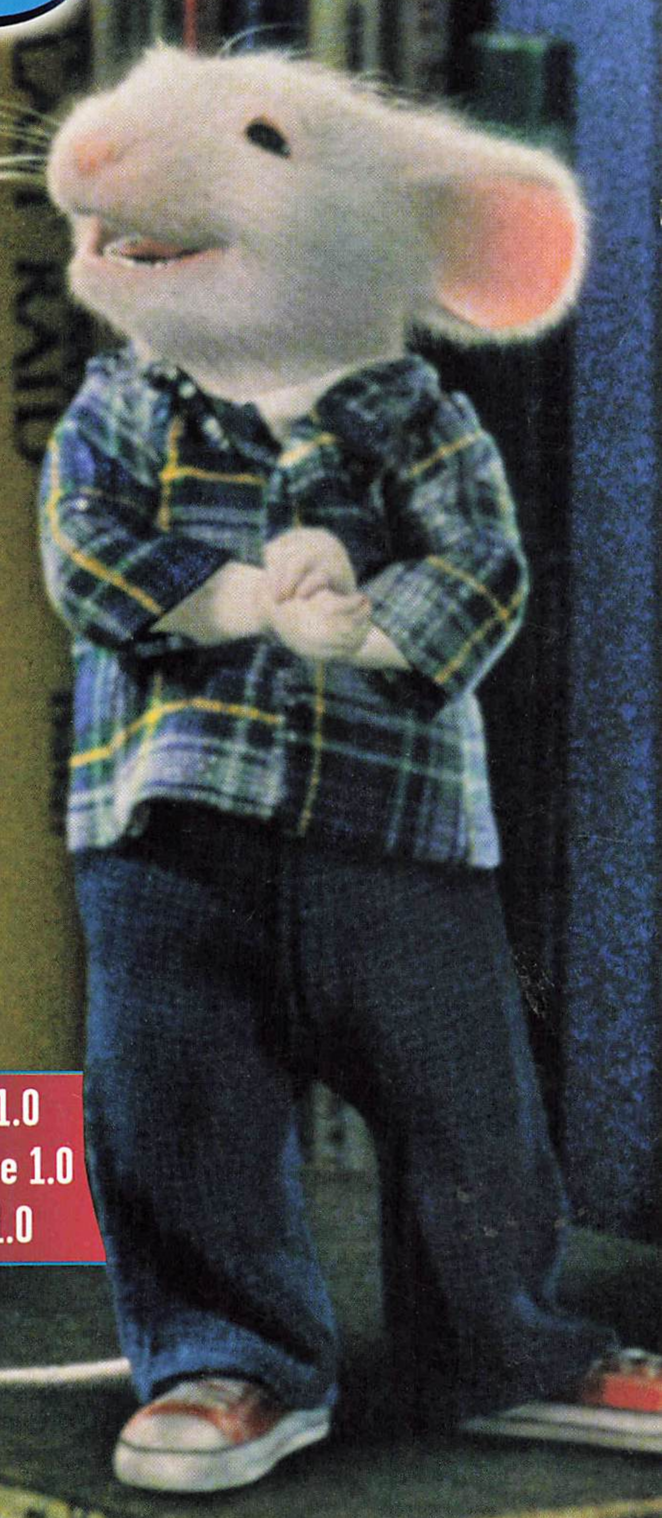
ADVANCED GRAPHICS & ANIMATION

FEBRUARY 2000

3D



No Animals Were Harmed
in the Making of This Movie

- Stuart Little's Digital Fur
- Hair & Fur Tools Roundup
- Fur Rendering at ILM

TUTORIAL

Customizing the MAX Interface

IN-DEPTH REVIEWS
- Nichimen Mirai 1.0
- Virtus OpenSpace 1.0
- NeMo Creation 1.0

www.3Dgate.com
Vol. 6, No. 2
U.S. $3.95 Ca

A MILLER FREEMAN PUBLICATION


IT COULD BE THE DIFFERENCE BETWEEN A POOR STARVING ARTIST, AND A RICH FAT ARTIST.

Dual Flat Panel Monitors Sold as an Upgrade.

NEW DELL® *PRECISION*™ Workstation 420

- Intel® Pentium® III Processor at 600MHz (up to 733MHz)
- Dual Processor/RAID Capable
- 128MB PC800 RDRAM at 400MHz (up to 1GB)
- 9GB[4] 7200 RPM Ultra 160/m SCSI HD (up to 36GB[4] 10K)
- 17" (16.0" vis) P780 Trinitron® Monitor (up to Flat Panel)
- Diamond Fire GL1 Graphics Card
- Integrated NIC & Sound; 48X Max[16] Variable CD-ROM
- Microsoft® Windows NT® Workstation 4.0; 1-Year NT Support
- 3-Year Next-Business-Day On-site[3] Service
- 7X24 Dedicated Hardware Phone and Online Tech Support

$3989 Business Lease[15]: $134/Mo.,36 Mos. E•VALUE CODE: **89492-490139**

- Upgrade to the *NEW* Intense 3D Wildcat 4110 Pro Graphics Card, add $1900

WITH UP TO DUAL INTEL® PENTIUM® III PROCESSORS AND THE EYE-BLOWING GRAPHICS OF THIS DELL® PRECISION™ WORKSTATION LINE, YOU'LL BE EATING STEAK IN NO TIME AT ALL. THAT'S BECAUSE THE EXTREME PERFORMANCE OF THIS WORKSTATION CAN GIVE YOUR ARTISTIC VISION (EXECUTED ON APPS LIKE ADOBE AND SOFTIMAGE®) A DIGITAL 3D POWER THAT WILL IMPRESS YOUR CLIENTS, AND TERRIFY YOUR COMPETITION. THERE ARE PLENTY OF STRUGGLING ARTISTS OUT THERE. MAYBE YOU CAN LOAN THEM SOME MONEY?

DELL® *PRECISION*™ Workstation 610

- Up to Dual Intel® Pentium® III Xeon™ Processors at 550MHz (RAID Capable)
- 64MB ECC SDRAM (up to 2GB)
- 9GB[4] up to 36GB[4] (10,000 RPM) Ultra-2/LVD SCSI HDs
- 32MB Diamond Viper V770D Graphics Card (Upgradeable to: 32MB Matrox G400 Max, Appian Jeronimo Pro, Diamond Fire GL1 or Intense 3D Wildcat 4000)

STARTING AT

$3129 Business Lease[15]: $105/Mo.,36 Mos. E•VALUE CODE: **89492-490131**

NEW DELL® *PRECISION*™ Workstation 220

- Up to Dual Intel® Pentium® III Processors from 533MHz-733MHz
- 64MB PC800 RDRAM at 400MHz (up to 512MB)
- 6.4GB[4] EIDE HD up to 27.2GB[4] EIDE or 9GB[4] up to 36GB[4] (10,000 RPM) Ultra-2/LVD SCSI HDs
- 32MB 4X AGP Diamond Viper V770D Graphics Card (Upgradeable to: 32MB 4X AGP Matrox G400 Max or Appian Jeronimo Pro)

STARTING AT

$2129 Business Lease[15]: $71/Mo.,36 Mos. E•VALUE CODE: **89492-490121**

DELL® *PRECISION*™ Workstation 210

- Up to Dual Intel® Pentium® III Processors from 500MHz-700MHz
- 64MB ECC SDRAM (up to 512MB)
- 6.4GB[4] EIDE HD up to 27.2GB[4] EIDE or 9GB[4] up to 18GB[4] (10,000 RPM) Ultra-2/LVD SCSI HDs
- 32MB Diamond Viper V770D Graphics Card (Upgradeable to: 32MB Matrox G400 Max or Appian Jeronimo Pro)

STARTING AT

$1859 Business Lease[15]: $62/Mo.,36 Mos. E•VALUE CODE: **89492-490118**

Common Features:

- **17" (16.0" vis) M780 Monitor [Upgrades available from 17" (16.0" vis) P780 Trinitron® Monitor to 24" (22.5" vis) and Flat Panel]**
- **Multi-Monitor Capability ■ 48X Max[16] Variable EIDE CD-ROM Drive ■ Integrated SoundBlaster Pro Compatible Sound**
- **Integrated 3Com® 10/100 PCI TX NIC with Remote Wakeup ■ Remote Client Manageability Support via Wakeup on LAN (WuOL) Capable**
- **Microsoft® Windows NT® Workstation 4.0 ■ 3-Year Next-Business-Day On-site[3] Service ■ 7X24 Dedicated Telephone and Online Tech Support**

WWW.DELL.COM/SMALLBIZ/DCC ■ 1.877.688.3355

BE DIRECT™

USE THE POWER OF THE E•VALUE™ CODE. Match our latest technology with our latest prices. Enter the E•VALUE code online or give it to your sales rep over the phone. WWW.DELL.COM/EVALUE

www.dell.com

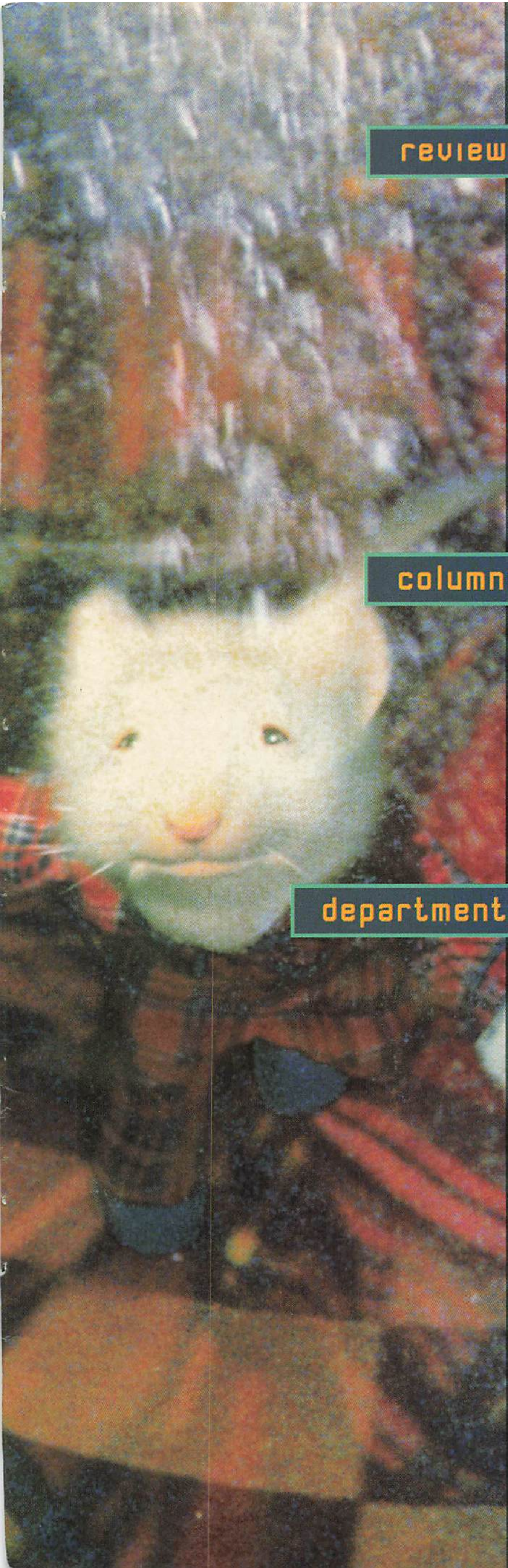
pentium® III

Phone Hours: M-F 7a-9p ■ Sat 10a-6p ■ Sun 12p-5p CT ■ In Canada[1], call 800-839-0148 ■ In Mexico[1], call 01-800-021-4531 ■ GSA Contract #GS-35F-4076D

Prices not discountable. [1]Prices and specifications valid in U.S. only and subject to change without notice. [2]For a complete copy of Guarantees or Limited Warranties, write Dell USA L.P., Attn: Warranties, One Dell Way, Round Rock, TX 78682. [3]On-site service may be provided by a third-party provider under contract with Dell, and is not available in certain areas. Technician will be dispatched if necessary following phone-based troubleshooting. [4]For hard drives, GB means 1 billion bytes; total accessible capacity varies depending on operating environment. [15]Business leasing arranged by Dell Financial Services L.P., an independent entity, to qualified customers. Above lease payments based on a 36-month lease, and do not include taxes, fees, shipping charges; subject to credit approval and availability. Above lease terms subject to change without notice. [16]20X Min. Intel, the Intel Inside logo and Pentium are registered trademarks; Pentium III Xeon and Celeron are trademarks of Intel Corporation. MS, Microsoft, BackOffice, IntelliMouse, Windows NT and Windows are registered trademarks of Microsoft Corporation. 3Com is a registered trademark and Fast EtherLink is a trademark of 3Com Corporation. Trinitron is a registered trademark of Sony Corporation.

3D February 2000

features



Cover and table of contents images courtesy of Sony Pictures Imageworks.

Forget megahertz.
We're talking gigaflops.

Presenting the Power Mac G4. The world's first desktop supercomputer.

What makes a supercomputer "super" is its ability to execute at least one billion floating-point operations per second. It is a staggering measure of speed known as a "gigaflop." The new Power Mac™ G4 is the first personal computer in the world to achieve this level of performance. The secret of this stunning speed is the new G4 processor with its Velocity Engine™—the heart of a supercomputer miniaturized onto a sliver of silicon. Applications that tap the Velocity Engine's power typically run up to twice as fast as they do on the fastest Pentium III-based PCs.* Common Photoshop tasks, for example, run twice as fast. And using a set of Intel's own tests, the 450MHz G4 chip was 2.3 times as fast as the 700MHz Pentium III processor. Chances are, you've never even heard of a gigaflop before. But very soon you won't be able to live without at least one on your desk. For more information and complete specifications, visit us at www.apple.com. Think different.™

What makes a supercomputer "super" is its ability to execute at least one billion floating-point operations per second. Like the new Power Mac G4.

The G4 chip incorporates the Velocity Engine—the heart of a supercomputer miniaturized onto a sliver of silicon.


www.3dwillnever
bethesame.com

3D

www.3Dgate.com

EDITOR-IN-CHIEF Ted Greenwald, *tgreenwald@mfi.com*
SENIOR EDITOR Gretchen J. Bay, *gbay@mfi.com*
TECHNICAL EDITOR Chris Tome, *ctome@mfi.com*
MANAGING EDITOR Matthew Hoover, *mhoover@mfi.com*
ASSISTANT EDITOR Mike Kobrin, *mkobrin@mfi.com*
ART DIRECTOR Audrey Welch
GROUP DIRECTOR/PUBLISHER Robert Melk, *415-278-5247, bmelk@mfi.com*
SALES DIRECTOR Debra A. Luft, *415-278-5266, dluft@mfi.com*
SENIOR ACCOUNT MANAGER Michelle Gorsuch, *415-278-5323, mgorsuch@mfi.com*
EASTERN REGIONAL SALES REPRESENTATIVE Kimberly Dusel, *415-278-5233, kdusel@mfi.com*
WESTERN REGIONAL SALES MANAGER Darrell Martin, *415-278-5219, dwmartin@mfi.com*
NORTHERN CALIFORNIA SALES REPRESENTATIVE Michelle Stutsman, *415-278-5270, mstutsman@mfi.com*
DIGITAL MEDIA MANAGER Chris Manners
ADVERTISING PRODUCTION MANAGER Sarah Breivis, *415-905-2537, sbreivis@mfi.com*
MARKETING DIRECTOR Kathleen Sachs
MARKETING COORDINATOR Jill Perretti
CIRCULATION MANAGER Henry Fung
CIRCULATION ASSISTANT Ron Escobar
CIRCULATION ASSISTANT Holly Nelson
NEWSSTAND MANAGER Pam Santoro
REPRINTS Sherry Bloom, *415-808-3980, sbloom@mfi.com*
GENERAL INFORMATION 415-905-2200

Miller Freeman
A United News & Media publication

CEO/MILLER FREEMAN WORLDWIDE Tony Tillin
CHAIRMAN/MILLER FREEMAN INC. Marshall W. Freeman
PRESIDENT Donald A. Pazour
EXECUTIVE VICE PRESIDENT/CFO Ed Pinedo
EXECUTIVE VICE PRESIDENTS H. Ted Bahr, Darrell Denny, Galen A. Poss, Regina Starr Ridley
SENIOR VICE PRESIDENTS Wini D. Ragus, John Pearson
SR. VICE PRESIDENT/PRODUCTION Andrew A. Mickus
VICE PRESIDENT/CIRCULATION Jerry M. Okabe
SENIOR VICE PRESIDENT/ CREATIVE TECHNOLOGIES GROUP KoAnn Vikoren
VICE PRESIDENT Johanna Kleppe

3D ADVISORY BOARD
Matt Elson, *X07*
Tim Forcade, *Forcade & Associates*
Barrett Fox, *infoplasm*
Patricia Johnson, *Industry Consultant*
Sanford Kennedy, *Sanford Kennedy Design, EleFont*
Isaac Kerlow, *The Walt Disney Company*
Jeffrey Lerer, *Independent Animator*
Alex Lindsay, *Visual Effects Artist*
Jim McCampbell, *Ringling School of Art & Design*
David Munier, *Fox Animation Studios*
Nick Napp, *Rainbow Studios*
Mark Thompson, *The Big Machine*
Scot Tumlin, *GameTek*
Sue Wilcox, *VRML Industry Analyst*

INTERNATIONAL LICENSING Abramson and Associates, *914-723-4700*

BPA INTERNATIONAL®

YOUR SATISFACTION MATTERS: Our customer-service staff is dedicated to making sure your subscription to *3D* is hassle-free. If you miss an issue, have a billing problem, want to renew your subscription, or just have a question: CALL US! (800) 234-4286 or (615) 377-3322; FAX US! (615) 377-0525; EMAIL US! custserv@cwixmail.com.

POSTMASTER: Send address changes to: *3D*, Circulation Dept., PO Box 5031, Brentwood, TN 37204-5301.

EDITORIAL POLICY 3D is an independent publication without obligation to any hardware or software vendor, dealer, or distributor. Articles marked as news or features are written without knowledgeable bias. Articles marked as opinion represent the opinion of the author. Contributors may be reimbursed for articles, depending on the nature and size of the submission. Send articles or editorial queries to *3D*, 525 Market St., Ste. 500, San Francisco, CA 94105.

Fahrenheit Hot & Cold

The one-two punch of the U.S. Justice Department's antitrust decision against Microsoft and SGI's recent reorganization has left a lot of people wondering about the fate of Fahrenheit. Announced in late 1997, Fahrenheit was an effort to make the world safe for graphics development at both high and low ends by integrating Direct3D, Microsoft's proprietary 3D graphics API, with OpenGL, an open standard sponsored by SGI.

You might think trouble with the Feds would prompt Gates & Co. to embrace OpenGL all the more tightly, if only to demonstrate that they aren't using their dominant position in operating systems to shut out competition. But a cache of email recently intercepted and released by the Register (www.theregister.co.uk), an online IT journal, suggests otherwise.

"OGL is not strategic for us," the Register quoted Philip Taylor, DirectX Evangelist at Microsoft, on November 29. "Softimage has about 20,000 seats total.... Outside of the *Quake* family of games there are, maybe, two hands-full of apps that use OGL.... D3D has overwhelming support in terms of titles, yet we have a serious lack of drivers. D3D drivers are strategic for us." The report concluded that OpenGL would not be included in Windows 2000.

On December 1, Taylor distributed an official Fahrenheit/OpenGL FAQ, insisting that he had been quoted out of context. "My email was a personal lament over general lack of drivers for W2K and not policy," he wrote. Windows 2000 will support OpenGL 1.1; OpenGL 1.2 will be added in a service pack shortly thereafter. Fahrenheit is alive and well, with plans to ship a final version within the first half of this year.

D3D dominates consumer 3D titles, it's true, but that dominance has never reflected a preference on the part of title developers. It was agitation by those very developers that spurred Microsoft to pursue Fahrenheit, and meanwhile, OGL has made tremendous strides in the consumer market. Historically, it has been more precise, stable, extensible, and backwards-compatible, and it supports curved surfaces and addresses more of the 3D graphics pipeline. Moreover, OGL has engendered truly professional tools for 3D designers, artists, and animators. They aren't perfect, but they outstrip anything D3D would have given us.

Given the uncertainty of Microsoft's commitment, developers of high-end OGL-based apps must choose either to spend an immense amount of R&D money to switch over to D3D, or to stick with a proven technology and the promise of low-cost UNIX boxes. OGL's heritage in IRIX means it can be supported readily on other UNIX-based platforms, including Linux and Mac OSX. While the gap between OGL apps and D3D apps was narrowing, professional users happily chose the less expensive alternative. But if it widens, top-tier professionals are likely to choose OGL, especially if it doesn't cost much more.

If Microsoft's strategists look at the professional content creation market and see only 20,000 seats of Softimage, they're missing the bigger picture. 3D graphics don't exist in a vacuum. Increasingly, digital media production demands an integrated platform. Even in the face of enormous setbacks, Apple has turned the Mac's early dominance in 2D and audio into long-term strength in video and web authoring. Conversely, despite the short-term success of Windows NT, Windows remains weak in not only 3D graphics but 2D, video, and audio as well.

In the era of the post-PC entertainment platform, D3D might lose in the consumer market as well. The Sony PlayStation 2 portends a new generation of high-performance graphics devices that won't be hobbled by D3D or any other general-purpose API. If the coming wave of set-top boxes and Internet appliances is successful, the API question will be a very different one.

With or without Fahrenheit, dropping OpenGL support would be a risky move for Microsoft. It would certainly make life more difficult for 3D designers, artists, and animators, not to mention hardware and software vendors. I hope the Department of Justice can persuade them not to do it.


POLICE
1

BY MATTHEW HOOVER

Napalm, Fahrenheit & Friends

3dfx updates Voodoo to 4 & 5, Fahrenheit may be alive, and the form•Z kids are alright.

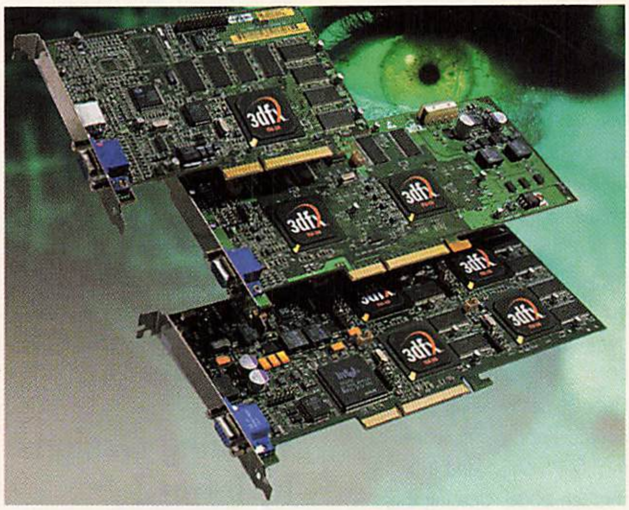
The Napalm is finally out, but that's a good thing. 3dfx's next-generation chipset, code-named Napalm, is the basis for both Voodoo4 and Voodoo5 boards, with the formal name Voodoo Scalable Architecture or VSA-100. The scalable part reflects that it's actually up to four VSA chips per card, cooperating via scan-line interleaving. This boosts the chip's polygon fill rate far beyond Voodoo3, and it does full-screen anti-aliasing, texture compression, motion blur, depth of field, and smooth reflections. The VSA chips support DirectX, GLide, and OpenGL, but only the higher-end configurations are designed to perform in the windowed environment of a professional 3D app.

Voodoo4 4500 ($179) is a single chip AGP or PCI board, but because it's a single chip, it won't be able to do the VSA's cool scan-line interleaving or anti-aliasing, which require two or more chips. The Voodoo4 4500 board will have 32MB VRAM and push about 350 million pixels per second.

The Voodoo5 5000 ($229 PCI) contains two chips per board, has 32MB VRAM, can render four pixels per clock cycle (Voodoo4 does two), and pushes about 700 million pixels per second. The Voodoo5 5500 ($299 AGP) has two chips and 64MB of VRAM. The Voodoo5 6000 ($599) has four VSA-100 chips for eight pixels per clock cycle, 128MB VRAM, and around 1.4 billion pixels per second.

3dfx has not announced a Mac-specific product, although PCI Voodoo4 and Voodoo5 will work with 3dfx's Mac drivers.

Speaking of drivers, 3dfx has released its GLide API spec and will focus its future efforts on OpenGL and Direct3D drivers. The company also made open-source its hardware specs for currently shipping 3D accelerators. This should give more 3D capabilities to Linux systems and perhaps encourage more companies to port 3D games to the Linux OS, not to mention give 3dfx an inroad to a market that's increasingly important as Voodoo's dominance of the PC gaming world is eroding. In August, 3dfx made its FXT1 texture compression technology open-source, as well.



The newest members of the 3dfx family, the VSA 100-based boards called Voodoo4 and Voodoo5.

Where's OpenGL in Windows 2K?

Although details remain murky, Microsoft is not shipping any OpenGL drivers in Windows 2000. The company plans to include "support for" OpenGL, but vendors will need to write their own drivers, according to Microsoft and IT zine the Register (www.theregister.co.uk).

At first, Win 2K beta testers noticed OpenGL apps performing below par or crashing, and Microsoft responded with a variety of answers, according to emails received by the Register. The company said support for some 3D cards had been dropped, it said all support was there, and eventually it said work won't begin on OpenGL til "Direct3D is 100 percent."

After the Register broke its story, Microsoft denied it wasn't supporting OpenGL but didn't refute all the inconsistencies in its communications. Microsoft also said Fahrenheit—its next-generation graphics API collaboration with SGI—was on track and the first piece, the Fahrenheit Scene Graph, will ship in the first half of 2000, this after months of silence and SGI's withdrawal from the project. Fahrenheit's roots go back a couple years to when Microsoft needed 3D graphics for Windows NT. SGI was developing its first NT-based workstations and needed cross-platform graphics. Today, with Microsoft in a stronger position and SGI in a weaker one, development in Redmond is focusing on Microsoft Direct 3D drivers and incorporating "OpenGL technology" into it. However this shakes out, it looks like straight-from-the-box Windows 2000 may need some help to run your high-end 3D apps.

More Changes to SGI's Roadmap

Moving further down the road to rebuilding itself, SGI announced another change to its strategy for Intel-based workstations. After saying it would find another manufacturer for its Visual Workstation line, then admitting it hadn't found one, the company said it is keeping Intel workstations in-house but drastically changing its NT product line.

The 320 and 540—long in development, late to arrive, stymied by mechanical delays and distribution problems—are history, to be replaced by a single desktop system. The new workstations will be available with Windows NT or Linux, and will be not be distributed directly by SGI, as was tried at first with the 320 and 540. Resellers eventually sold the machines, but the channel

wasn't well developed, and margins were very low, so plans changed again.

By now, with the 840 chipset, Intel has caught up to the data-transfer advantage that SGI was developing for its Intel debut. The new workstations will have a more industry-standard system architecture—no more specialty SGI hardware.

Windows NT is at least partially in the plans for a while, but the company reaffirmed its commitment to Linux—to the point of announcing some OpenGL software will be open-source—and plans to use its own MIPS chips and IRIX OS through 2006. Coming this year are the successors to the chips under the hood of the O2 and Octane workstations, the R5000 and R12000.

It's been a long, arduous, expensive foray into high-end desktop computing. SGI filed papers with the Securities and Exchange Commission stating they're out $40 million for cancellation of the 320 and 540 manufacturing contracts, plus $20 million for excess inventory. SGI is still looking for a buyer for the Cray Research supercomputer division.

Partnerships & Updates

Discreet announced it has shipped its 100,000th copy of 3D Studio MAX 3.0—shipped, not sold, but still an impressive figure. By the way, Maya 3.0 is scheduled to be announced at SIGGRAPH 2000 in New Orleans, the long-awaited LightWave 6.0 is still scheduled to get out the door in 1999, Strata Studio Pro's next rev is in beta, and Maxon Cinema 4D 6.0 isn't too far from market, either.

Softimage's successor, codenamed Sumatra, should be out of beta and into the 1.0 frying pan by late March. (Is Avid going to do the sensible thing and name it Sumatra, or do they really have a better name under wraps?) Avid announced in December that Sumatra has been used to complete its first commercial project, a title sequence for the BBC series "Everyman," created by Aldis Animation of London. Kim Aldis, founder of Aldis Animation, said Sumatra's scripting capabilities and real-time render previews were key to getting the job done.

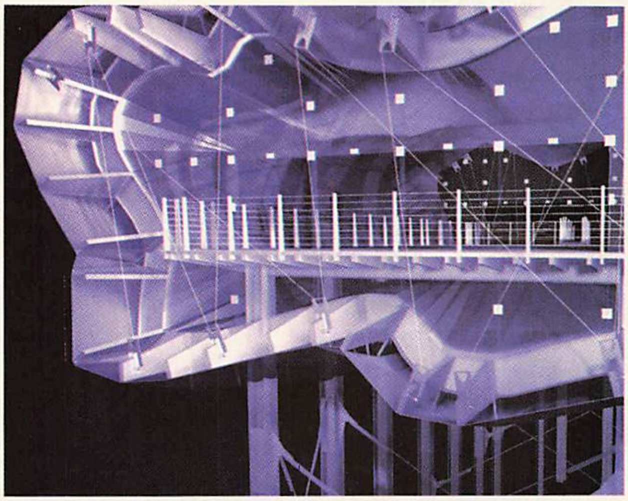
Winning Designs by form•Z

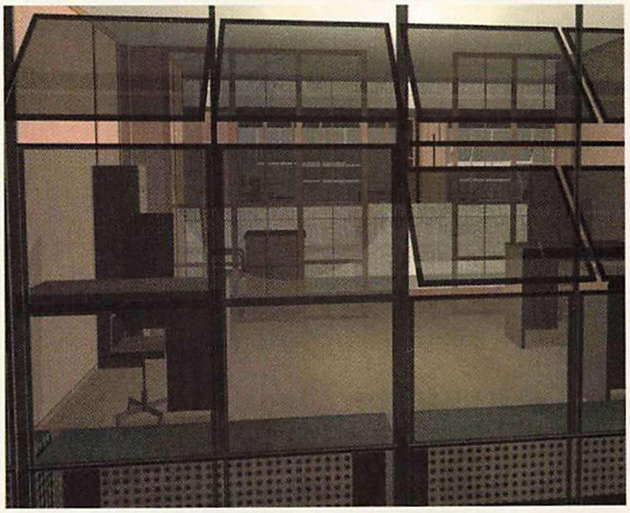
auto·des·sys announced the winners of its seventh annual Joint Study Program Awards, which honor students who use auto·des·sys form·Z modeling software to creating outstanding designs.

The program provides form·Z to more than 300 institutions—colleges, universities, and now a few high schools—who incorporate 3D modeling into their curriculum. The schools give their experiences with and recommendations about the software back to auto·des·sys.

This year four Awards of Distinction and 10 Honorable Mentions were granted in four categories: Architectural Design, Urban and Landscape Design, Product and Industrial Design, and Visualization/Illustration. A jury of six industry and academic experts selected the winners.

Tristan d'Estree Sterk of the University of Adelaide, Australia, took top honors in the architectural category with "Clouds," his design for a performance hall situated in the ocean adjacent to a beach—a challenging site to say the least, but his stylized designs fit the environs.





Detail of "Clouds," a performance hall by Tristan d'Estree Sterk (top), and interior shot of "Frank Lloyd Wright's National Life Insurance" by Dean DiSimone and Joseph Kosinski, both winners in auto•des•sys' annual Joint Study Program Awards.

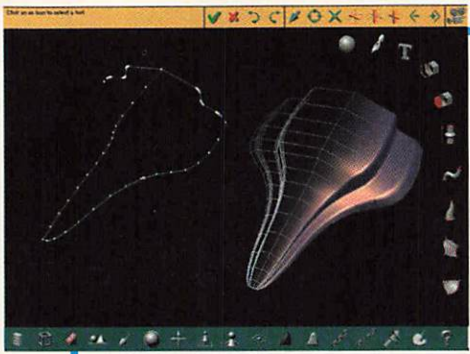
Damian Etrocassi of the Universidad de Buenos Aires, Argentina, won the industrial design category with his inventive computer desk entitled "Office Furniture," a very detailed, ergonomically designed workspace that addresses today's computer systems.

"The Meeting of Two Worlds," a festival and information building for the 2000 Olympics, by John Endersbee of the University of Adelaide, Australia, won the Award of Distinction in the urban design category. The project addresses how we build memorials and what sort of landscape is "authentically" Australian.

Dean DiSimone and Joseph Kosinski of Columbia University won the visualization category with their project, "Frank Lloyd Wright's National Life Insurance," which visualized a building Wright drew plans for but never built. In typical Wright fashion, the interior walls are suspended screens that seem to levitate. Congratulations to all winners on their stellar designs.

Matthew Hoover is managing editor and news editor for *3D*. Send your news about products, projects, and technology to mhoover@mfi.com.

BY MATTHEW HOOVER



Amapi 3D 5.0 & Tutorial CD

➤ TGS Software released Amapi 3D 5.0 ($399), its Windows and Mac 3D modeling software that features a new method of editing, plus new tools for smoothing, deforming, and filleting models. Amapi 3D 5.0 offers NURBS and polygonal curves and surfaces for modeling, and it tracks the construction history of a model so you can edit it based on its orginal outline, profile, or structure, which TGS calls Dynamic Geometry. The program includes a new shaded wireframe viewing mode. Amapi 3D comes with four new methods of smoothing meshes, a new filleting tool palette that can fillet all corners you select to a numeric value, and tools for bending, twisting, and tapering your models via Meta-NURBS. Amapi 3D 5.0 adds Maxon Cinema 4D to its long list of export formats, which includes 3D Studio MAX, LightWave, Softimage, Ray Dream Studio, trueSpace, and NeMo, as well as STL and ZAP. With ZAP, you can export complex 3D scenes for delivery over the web to a computer that has the free client player. Amapi 3D 5.0 is available for Mac and Windows 95/98/NT. **www.tgs.com**

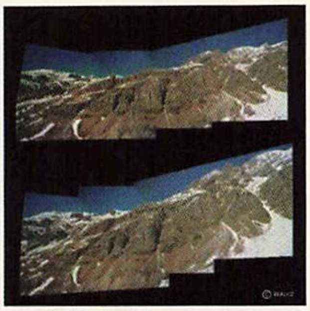
Also available is the Animhouse Amapi training CD ($35). This interactive QuickTime movie incorporates video of hands, a keyboard, and a screen in an Amapi interface as it walks you through modeling and surfacing techniques and shows off many time-saving tips. **www.animhouse.com**

dpsReality Studio Disk Recorder

➤ Digital Processing Systems (DPS) is now shipping dpsReality ($2,995), a studio digital disk recorder designed for animators, 2D artists, and broadcast designers. dpsReality includes software for previewing clips, overlaying graphics, and controlling decks. It ships with Digital Fusion DFX, a special version of eyeon software's compositing tools modified to support the dpsReality hardware. One new feature is the Virtual Tape File System, which allows simultaneous network access from multiple machines, so one user can render to the system while it plays video in real time. Hardware features include onboard disk control and video I/O, mixed-mode compressed/uncompressed video and alpha channel storage and playback, four-track audio, and real-time VGA preview. It can be expanded with a 3D video daughterboard and real-time DV I/O. **www.dps.com**

REALVIZ Stitcher

➤ REALVIZ Stitcher ($2,000, or pay-per-use license) for Windows creates panoramic images from photographs. Scanned images are imported and manually roughed out, then Stitcher automatically warps and stitches them, blending colors and deblurring as necessary.



Images up to 360x360 degrees can be created and exported to 2D compositors for matte painting or 3D packages for mapping. Windows NT and IRIX versions are scheduled to ship during Q1 2000. **www.realviz.com**

FAMOUSfaces 1.5

➤ FAMOUS Technologies has released FAMOUSfaces 1.5 ($4,990), a standalone and plug-in facial animation system for Windows NT that now supports real-time game animation and multiple inputs (mocap, video, puppetry, voice recognition, cluster targeting, or a combination). FAMOUSfaces can extract 3D facial motion from AVI files and NTSC video, and has mocap support for systems from Motion Analysis, Phoenix, Vicon, and Xist. It plugs into LightWave, Softimage, 3D Studio MAX, and Maya, and supports Kaydara Filmbox. **www.famoustech.com**

SceneGenie MAX Plug-In

➤ Autonomous Effects is shipping SceneGenie ($995), its plug-in suite for 3D Studio MAX 2.5 and 3.0. SceneGenie has more than 50 components and can match, track, and control cameras, colors, lights, paths, and more. It does color and lighting correction and matching and helps you animate complex models and paths without writing complex formulas. You can use IK on motion created by modifiers or animated morphs, and you can reconform mocap data to drive a different structure. Also available is SceneGenie Limited ($495), which has the same functionality as the full version but without sensors, visors, UV/HSV/Moment comparators, color/outline Waldos, visor material, and visor saver objects. Both versions are compatible with MAX 2.5 and 3.0. **www.afx.com**

Labtec Spaceball 4000 FLX

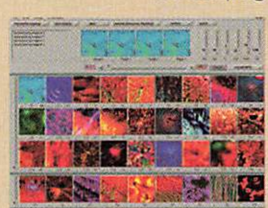
Labtec introduced the Spaceball 4000 FLX 3D motion controller ($695), an input device with a movable rubber ball, buttons, and ergonomic features to reduce hand and wrist strain. The Spaceball's nine finger buttons give instant access to camera and object movement. You push, pull, and twist the central PowerSensor ball to translate in X, Y, and Z. You can move both camera and object simultaneously, which can greatly speed up workflow. It includes customizable configurations, easy switching between right- and left-handed setup, and adjustable sensitivity. See Labtec's web site for a list of supported 3D and CAD applications. Spaceball is supported by apps on UNIX and Windows 95/98/NT. **www.labtec.com**

Free ParticleSuite for Softimage 3D

Registered users of Softimage 3D 3.8 can download Phoenix Tools ParticleSuite for free from www.softimage.com. (Phoenix Tools will provide cloth and dynamics modules for Sumatra, Softimage 3D's successor, scheduled to ship Q1 2000.) This particle generation system includes the ability to make any scene object a particle source, view any deformation in the Softimage user interface, and simulate fluids and gasses, plus it includes mental ray shaders. Valid Softimage 3D 3.8 SP2 or later registration and "extreme" level service contract required. **www.phoenixtools.com, www.softimage.com**

Better Texturing Through Genetics

Cinema Graphics GenShade ($5,000 per floating license) is a shader algorithm app for creating Pixar RenderMan and Side Effects Houdini shaders on IRIX. GenShade is based on genetic algorithms, so it produces a population of shaders in a parent-child fashion from pre-existing shaders and generation parameters. It displays a thumbnailed selection of shaders you can tweak and refine. GenShade combines shaders to produce "offspring" shaders interactively without programming. GenShade can import shaders in Cinema-Graphics ShadeTree format and can export files for RenderMan, Houdini, or ShadeTree renderers. GenShade supports surface, displacement, light, and atmosphere shaders, and it comes with several prebuilt example shaders. **www.cinegrfx.com**



Feel the GeForce

ELSA announced its new Erazor X line of graphics accelerators based on the nVidia GeForce 256 chip, aimed at players of DirectX–based PC video games. The Erazor ($269) has up to 32MB of memory, 32-bit color depth, and resolutions up to 1920x1280. The Erazor X line supports T&L, the new buzzword in graphics cards, which means transform and lighting. ELSA's Revelator stereo shutter glasses ($69) also support T&L. The ELSA GLoria II is a 64MB OpenGL card based on the GeForce. **www.elsa.com**

stray pixels

Dosch Design of Germany has released three new CDs of textures for Maxon Cinema 4D. Rust and Metal, Stone and Concrete, and Skin are $59 each and include more than 100 1024x1024 seamless textures with accompanying shader maps (such as color, bump, and transparency). **www.doschdesign.com, www.maxon.net**

Digital Forays has released its first product, the 3D Forest CD-ROM ($99), which contains 290 3D tree models in 3DS and DXF formats (80 woody trees, 75 palms, 65 conifers, 50 shrubs, and 20 dead trees). On the company's web site are free textures for the models, currently in MetaCreations Bryce 4.0 format, but soon to include Caligari trueSpace and NewTek LightWave/Inspire. Additional models are also available at **www.3dforest.com**

Exabyte Corp. announced its Mammoth-2 tape drive ($4,895) for storage and backup. It can transfer at 12MB/sec onto 60GB cartridges with built-in cleaning tape, which cleans it automatically. The Mammoth-2 is automation-ready and backward-compatible with previous Mammoth tape libraries. Works with most major software and all major OSs, including Linux. **www.m2wins.com**

Able Software 3D-Doctor 3.0 ($2,400) is medical imaging software that creates 3D surface models and volume rendering from 2D cross-section images in minutes on low-cost PCs. Users can export 3D polygon models to DXF, 3DS, STL, VRML, and other formats. 3D-Doctor also features the 3D Basic scripting language for extending functionality. **www.ablesw.com**

SupremeGS is going all-Linux. The manufacturer is replacing its MediaStorm BeOS and HyperStorm Linux boxes with the MediaStorm II line of workstations, which will be exclusively Linux. Exact specs are still being finalized but will feature improved audio and video hardware. **www.supremegs.com**

S3 announced the Diamond Monster Shades ($69) stereoscopic shutter glasses for stereoscopic viewing of video games. They connect to the VGA output of a PC's video card (cabled and infrared versions available), are compatible with any OpenGL or Direct3D, and require a CRT monitor capable of 120MHz refresh rate. **www.s3.com**

IntelliStation M Pro
with Windows NT®

price	speed	creative
from $2,139* monitor not included	up to two Pentium® III processors 600mhz / up to 1gb ram IBM® Fire GL1† / Intense 3D graphics	You're in the hot seat. It's time to create. Time to direct. Time you don't have. Unless you have the IntelliStation M Pro. Load it with accelerated graphics solutions like the 25 bit IBM Fire GL1 graphics accelerator, and you'll have time to do everything. www.ibm.com/intellistation or call 800 426 7255, code is150

e-business tools

creative.director

IBM®

M Pro Options by IBM

memory Up to 128mb 100mhz ECC sdram. Run applications faster.

speed Add a second 500, 550 or 600mhz Intel® Pentium III processor. Work even faster.

media A read-write cd drive lets you write and transport 650mb of data.

intel inside pentium® III

I ♥ My Toolbar

by MICHELE BOUSQUET

Work faster and funner in Discreet 3D Studio MAX 3.0 with a custom GUI, including your own toolbars and buttons.

One of Discreet 3D Studio MAX 3.0's best new features is the customizable, floatable, do-anything-you-want-with-it toolbar setup. All your favorite tools can be placed right at your fingertips for quick access to just about all of MAX's functions.

Once you've placed your custom toolbars, you can save the customized GUI and load it anytime. It's sort of like having your very own custom-built version of MAX.

Default GUI When MAX 3.0 is loaded for the first time, the default GUI is displayed. Along with the familiar menus and the command panel at the right of the screen, there is a series of tab panels just under the menus, as shown in Figure 1.

By default, the tab panel called Main Toolbar is displayed. This tab is a replica of the solo toolbar from 2.5. All the familiar buttons are there in 3.0, and aside from a few new flyouts and features, all perform the same functions as the previous version's tools.

The difference in 3.0 is that the Main Toolbar is just one of many choices for tool display. When each tab is clicked, a set of tool buttons appears, with each button corresponding to a function on a command panel. For example, when you click the Lights & Cameras tab, the button display changes to a series of new buttons such as Target Spot, Target Direct, and Free Camera.

While the default tab panels are useful, the real deal with tab panels is their customizability. Tab panels are actually toolbars that just happen to be hanging out at the top of the screen. A toolbar is a display of tools that can be moved anywhere on the screen, staying there until you specifically tell it to go away, like the toolbar in Adobe Photoshop. You can create any number of new toolbars and/or tab panels with buttons and features you choose yourself, then turn it into a free-floating toolbar quicker than you can say "Customizable GUI."

Abracadabra, You're a Toolbar

To change a tab panel to a toolbar, simply click on the tab name and drag it into a viewport. The tab panel disappears and a toolbar appears instead, floating in the spot you dragged it to and displaying all the same buttons as the tab panel. In place of the tab name, a toolbar has a title bar that extends all the way across the top of the toolbar.

To move a toolbar to another spot on the screen, just click and drag its title bar. You can also get more choices by right-clicking on the title for a pop-up menu (Figure 2). The Dock option parks the toolbar at the top, bottom, left, or right of the viewport area. You can also dock a toolbar by dragging it to the edge of a viewport area. The shape of the cursor changes as you drag the toolbar, displaying the toolbar as a horizontal or vertical bar depending on the area to which it's being dragged.

Once a toolbar is docked, its title is collapsed into a tiny vertical or horizontal space at the end of the toolbar. To float the toolbar again, you'll need to right-click in this collapsed area to access the same pop-up menu again, where the Float option can be selected.

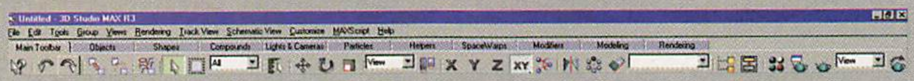


FIGURE 1. New tab panels are part of the default GUI in 3D Studio MAX 3.0.

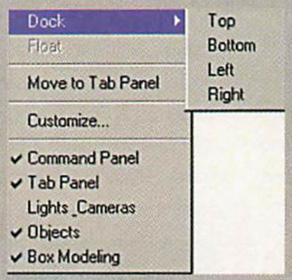


FIGURE 2. Right-click on a toolbar title to access more options.

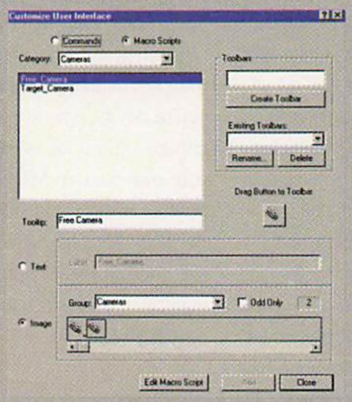


FIGURE 3. The Customize User Interface dialog lets you add any function to your new toolbar.

No More More Over time, MAX users develop their own familiar and surefire procedures that they can count on to get the best results in the shortest period of time. As an example, we'll consider the box-modeling approach. This strategy starts with a box or other primitive object such as a Gengon. Using a variety of modifiers, the object's vertices and faces are molded into a rough shape. Apply the MeshSmooth modifier, and voilà! A beautifully smoothed object.

Many MAX users find the box-modeling process efficient and effective for characters and organic objects. The only drawback is the numerous time-consuming trips to the Modify panel. Box modeling makes frequent use of a dozen or so modifiers such as Edit Mesh, Face Extrude, and Mesh Select. MAX 3.0 comes with about 80 different modifiers, and clicking the More button to search through the list gets old quickly.

You can remedy this by using the Configure Button Sets feature on the Modify panel to display all the necessary modifiers, but this creates another problem. Every modifier displayed takes up valuable space on the Modify panel, meaning you'll spend a lot of time scrolling down the panel to get at the parameters. Wouldn't it be great if you could close up the Modifier rollout altogether and use all your real estate for modifier parameters?

This is where a custom toolbar comes to the rescue. Here, we'll create a custom toolbar with all our favorite box-modeling modifiers, then dock it to the right of the viewport area.

A Toolbar I Can Call My Own

To create a custom toolbar, go to the Customize menu and choose Customize UI. You can also right-click on any existing tab panel or toolbar and choose Customize from the pop-up menu. Either way, the Customize User Interface dialog appears (Figure 3).

This dialog allows you to associate a particular MAX operation with a button picture of your choosing. Drag the button picture to the toolbar to create a shortcut to the operation.

Before creating the toolbar, you have to name it. Enter a name just above the Create Toolbar button. For our example, we'll call the toolbar Box Modeling. Click the Create Toolbar button to start the process. A small empty toolbar appears to the left of the dialog, floating over the viewport display.

We need to make some room in the toolbar to place the buttons. Move the cursor over the rightmost end of the new toolbar until the double-headed arrow appears, and click and drag to the right to extend the toolbar size by an inch or two.

Now for the buttons. From the Category pulldown at the top of the dialog, choose Modifiers. All the MAX modifiers appear in the area below the Category, listed in alphabetical order.

Click on a few different modifiers to highlight them. When you select a modifier, a series of picture buttons appears near the bottom of the dialog. One of the picture buttons has a thin black border around it. This is the suggested button picture for the selected function. The button images are arranged in groups of related modifiers. For example, if you highlight Edit Mesh then Edit Patch, you will see that the image series is the same, but a different button is outlined.

Highlight the Edit Mesh modifier on the list, and click on various button images near the bottom of the dialog. As you click on each image, you will see a copy of the image appear under the Drag Button to Toolbar heading at the right of the dialog.

Locate the button image that best communicates "Edit Mesh" to you, and click on it. Under the Drag Button to Toolbar heading, click and drag the button image to the new Box Modeling toolbar. The button appears on the toolbar.

Let's test the new button. Click Close on the Customize User Interface dialog. Create a primitive object on the screen and select it. Move the cursor over the new button. A tooltip should appear, telling you that this button represents the Edit Mesh modifier. Click the button. The command panel changes to display the Modify panel, and the Edit Mesh modifier is applied to the selected object.

To continue adding buttons to the toolbar, open the Customize User Interface dialog

again. Next, look for Mesh Select on the modifier list and highlight it. The same set of images is shown at the bottom of the dialog.

Suppose you aren't satisfied with the images displayed and want to see if there's anything else available that might more clearly state "Mesh Select." Just above the button series is a Group pulldown. Click the pulldown area and select Patches to see a new button series. The first button looks like a selection of faces and will work well as a Mesh Select button. Under Drag Button to Toolbar, drag the button to the Box Modeling toolbar.

Note that the dialog has two distinct areas: one for selecting an operation and another for choosing a picture button. You have to have the correct operation highlighted on the list at the upper left of the dialog. Once that's in place, you can choose any picture button you want to represent the operation. The selections on the Group pulldown display the images used on the default tab panels and are presented merely as suggestions for your custom dialog.

You can also change the name of the tooltip that will appear on your toolbar. As an example, let's use the Cap Holes modifier. Suppose that during box modeling, you usually use Cap Holes to make the mesh solid after you've cut a hole in it. For this reason, you'd like to have a tooltip that reads Make Solid for this modifier.

To do this, select the Cap Holes modifier from the list. The suggested picture button appears under Drag Button to Toolbar. In the Tooltip entry area, enter the title "Make Solid." Drag the button to the toolbar. Close the dialog, and move the cursor over the new button. The tooltip Make Solid should appear.

Open the dialog and highlight the MeshSmooth modifier. Suppose you can't find any picture buttons that satisfy you, and you'd like this particular modifier to appear as a label button with the name of the modifier spelled out. Choose the Text option. Under Drag Button to Toolbar, a label button appears with the name MeshSmooth. You can change the text on the button by entering new text in the Label entry area. Click and drag the button to the toolbar to create the new label button.

To complete our custom set of box-modeling modifiers, add buttons to the toolbar for the Mesh Select, Slice, Taper, and Relax modifiers.

Although we often refer to modifiers as commands, they are technically ASCII scripts built into MAX. At the top of the dialog, the Macro Scripts option is selected by default, and all the operation types listed on the Category pulldown are actually macro scripts. When an operation is selected from the list, you can click Edit Macro Script to see the ASCII script that performs the function.

Editing a macro script is not recommended unless you're really sure of what you're doing. However, viewing these scripts can give you a lot of insight into how to write your own custom MAX scripts. If you write a custom script and place it in the UI/SCRIPTS folder, it will appear on the list when Macro Scripts is selected, allowing you to assign a button to it and put it on a toolbar.

Command & Conquer Choose the Command option at the top of the dialog. Here, you have access to other types of commands, mostly operations from the Main Toolbar and certain menu options. In general, if you can't find the command you're looking for under Macro Scripts, you'll find it here under Commands.

When box modeling, you'll want to have some command functions handy. Scroll down the list and highlight Select Mode. Note that the picture buttons at the bottom of the dialog don't change. For commands, MAX doesn't present any button suggestions—you have to look through the available buttons and choose one that makes sense.

For commands from the Main Toolbar such as Select Mode, you can find the appropriate icons by choosing Main Toolbar from the Group pulldown. The Classic series displays button images from the MAX 2.5 Main Toolbar, which look slightly different from their 3.0 counterparts.

Choose Main Toolbar from the Group pulldown, and highlight the button that displays the selection cursor. Add this button to the toolbar. Next, choose Selection Floater from the functions list, and choose the button image that represents Select by Name on the Main Toolbar. Add this button to the Box Modeling toolbar.

Missing a Button? Sometimes you just won't be able to find a picture button that satisfies your tastes. In these cases, you can use your own custom-created button pictures. Custom button images have the advantage of making your screen unique and giving you a chance to express your personality. If you make a wide variety of weird buttons, you might even scare other users away and keep them from using your computer when you're at lunch.

To use custom buttons, you first have to create the button images with Photoshop or another paint program. Create a new 24x24 image, and draw or cut-n-paste to make the button image. Save the image in the UI folder as a BMP file with the name XXX_24I.BMP, with XXX as a meaningful filename. Reduce the image size to 16x16 and save another image file with the name XXX_16I.BMP. This will give you both a small and a large icon for use in MAX.

FIGURE 4. Custom buttons and transparency channels can be created in any paint program.

You'll need to close MAX and open it up again to have access to the new button images. Open the Customize User Interface dialog and look under Groups. The name you substituted for XXX should appear at the bottom of the list. Select it to make the button appear, associate it with any command or macro script, and put it on a toolbar.

Using a transparent background on a button image will make it look like the buttons that come with MAX. To make this happen, generate separate alpha channel images for each button image and save them with the filenames XXX_16A.BMP and XXX_24A.BMP in the UI folder. If your paint program doesn't support alpha channels, you can easily fake this effect by creating a separate grayscale image for each button with black in transparent areas, white in solid areas, and graduated grays for anti-aliasing (Figure 4). Save these images using the previous filenames.

You can also create groups of button images to appear in a row on the dialog. Each group of buttons is actually one image file which MAX automatically cuts up into 16x16 or 24x24 buttons. Each image must

sources

3D Studio MAX 3.0 • list price $3,495
Discreet
(800) 879-4233
www.discreet.com

be created with a height of 16 or 24 and a width that is a multiple of 16 or 24. To get an idea of how one of these images should look, load one of the existing group images from the UI directory in a paint program. These images all have filenames that end in _24I.BMP and _16I.BMP.

Location, Location, Location

Once your custom toolbar is complete, close the Customize User Interface dialog and move the toolbar around the screen. As you move it near the edges of the viewport area, the display outline will change shape, going from a horizontal bar near the top and bottom of the screen to a vertical bar near the left and right edges. Move the toolbar near the command panel at the right of the screen, and release the cursor. The custom toolbar is now docked as a vertical bar right alongside the command panel.

Because the MeshSmooth button has a text label, the toolbar takes up a lot of room horizontally. To make the toolbar take up less space, we'll change the text label to a button picture. Right-click the MeshSmooth label button and choose Edit Button Appearance. The Edit Macro Button dialog appears. This dialog looks very similar to the lower half of the Customize User Interface dialog.

Select the Image Button option, and choose Standard_Modifiers from the Group pulldown. Select any button image and click OK. You now have a fully functional toolbar for all your box modeling needs.

Clearing Up Some Space

A custom toolbar is the ideal companion for all the right-click viewport options available in MAX 3.0. Once you make the initial object for box modeling, you won't need access to the Modify panel very often. After you click your fabulous new Edit Mesh button, for example, you can access the vertex sub-object level by simply right-clicking on the object and choosing Sub-Object/Vertex from the pop-up menu. To make more room on the screen, you can hide the entire command panel by right-clicking on any of the blank areas of the menus, tab panels, or toolbars, and choosing Command panel from the pop-up menu to toggle it off. To give you an even larger work area, you can also hide the tab panel in the same way. This new arrangement of the screen is shown in Figure 5.

For those times when you do need the Modify panel, you can display the Command Panel again, but close up the Modifiers rollout to give yourself quick access to more of the current modifier's parameters.

Working with Toolbars

All toolbars can be hidden and re-displayed by right-clicking in the blank area of a menu, tab panel or toolbar, and choosing the toolbar name from the pop-up menu.

To hide a floating toolbar directly, click the X at the upper right corner of the toolbar title area. To hide a docked toolbar, right-click on the thin double lines at the end of the toolbar. Choose the toolbar name from the bottom of the pop-up menu to toggle it off.

If you prefer working with tab panels over toolbars, you can make the toolbar a tab panel in its own right. Right-click on the title bar of a floating toolbar or on the thin double lines at the end of a docked toolbar. Choose Move to Tab Panel from the pop-up menu. The toolbar is placed at the rightmost end of the tab panel display. To move it to another location, right-click on the tab panel name, and choose Move Left. Do this repeatedly until the toolbar is placed where you want it. To convert the tab panel back to a toolbar, click and drag on the tab panel name and drag it to the viewport area.

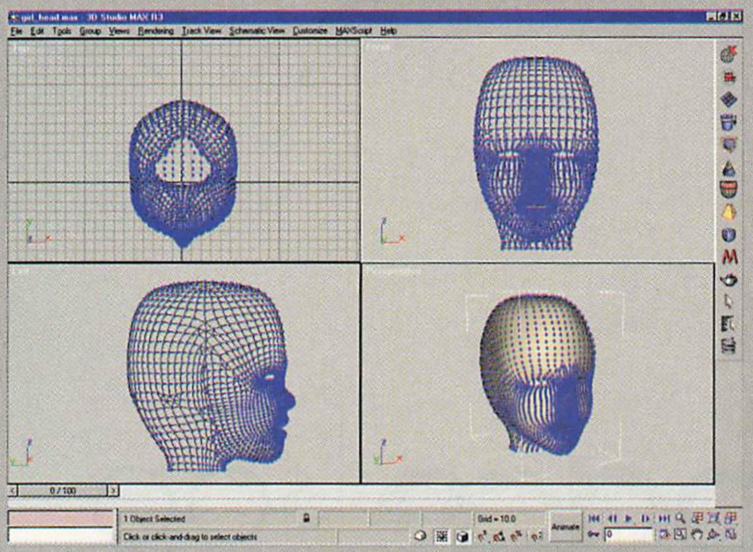


FIGURE 5. Wide, open spaces all ready for box modeling.

Saving the Custom UI

Once you set up the toolbars and other screen elements the way you like them, they'll stay that way until you say otherwise, even if you exit MAX and reload. If you share your workstation with other users who haven't acquired your same fine taste in toolbars, you can each have your own custom UI. To save the UI, go to the Customize menu and choose Save Custom UI As. Enter a filename on the file selector dialog and click Save. Your custom UI will be saved with the extension .CUI in the UI folder. You can even put the .CUI file on a floppy disk and carry it with you at all times for those emergency MAX situations.

Are Toolbars Right for Me?

Even though the default tab panels and toolbars are pretty cool, it's not a time-saver for all operations. Some functions, such as the PArray particle system, are difficult to communicate on a picture button. Although the MAX development team did a great job of making little pictures for big functions, it's sometimes hard to pick out the right one the first time you go to a tab panel. If you're looking for a feature you don't use very often, you'll probably find it quicker and easier to use the command panel.

For those features you use frequently, however, the tab panels and toolbars can make your work a breeze. Sprinkle your screen with custom buttons and express yourself through creative toolbar management. You'll be glad you did.

Michele Bousquet is a 3D Studio MAX artist and instructor based in New Hampshire. Her most recent project is a video on Discreet Character Studio 2.2. Visit her web site at www.maxhelp.com.

MicroScribe-3D™

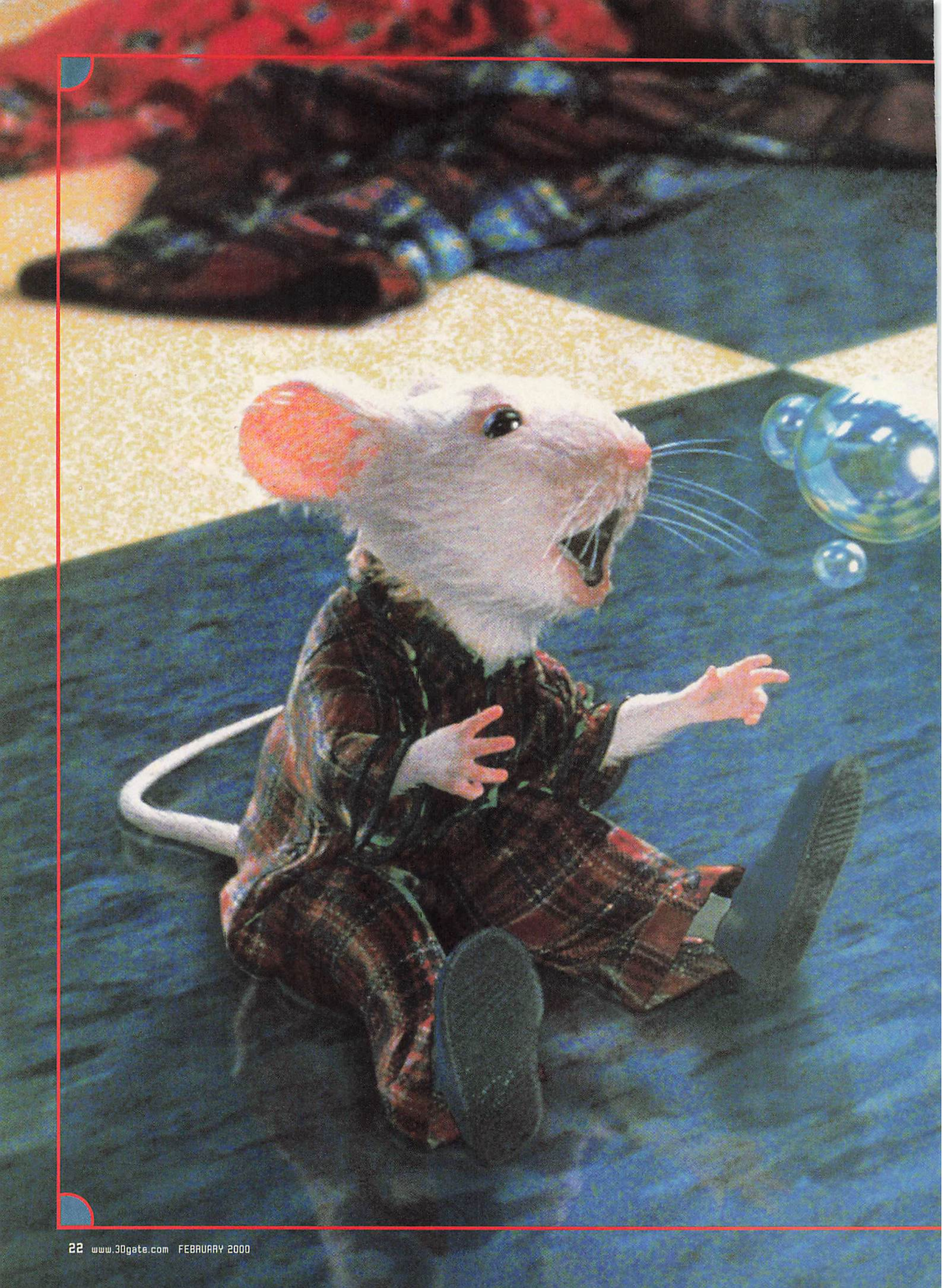
Desktop Digitizing Systems

Immersion

MicroScribe 3D is an advanced 3D digitizing tool. MicroScribe empowers you with the ability to capture complex 3D models from physical objects by simply tracing over their surfaces. The 3D digitizing process is fast, accurate, and more affordable than you might think!

The MicroScribe 3D is a desktop system that easily connects to your serial port and runs on PC, Mac, and SGI platforms. MicroScribe works directly in your favorite 3D graphics and CAD packages such as 3D Studio Max, Alias, Lightwave, Maya, AutoCAD, SolidWorks, form Z, Rhino 3D, and many more. MicroScribe comes in several size and accuracy configurations to me the demands of your project.

For details, call Immersion at 408-467-1900 or 1-888-467-1900 or visit our web site at www.microscribe.com



by BARRETT FOX

Sony Pictures Imageworks Builds a Better Mouse

The making of Stuart Little

At first glance, Sony Pictures Imageworks (SPI) seems hell-bent on exercising extremes. SPI produced high-impact effects for the 1998 remake of *Godzilla*, helping to bring to life a spectacularly huge and ferocious virtual creature. Now, this state-of-the-art special effects house has used its penchant for creation to conceive a character less than six inches tall and thoroughly steeped in cuteness.

Stuart Little is a completely computer-generated, bipedal, anthropomorphic, talking mouse. Originally the star of books penned by E. B. White, he is now the subject of his own Hollywood special effects blockbuster. SPI assembled a team of formidable talent for Stuart's genesis, many of them *Godzilla* veterans. To supervise the effects, they called on Academy Award-winning special effects producer John Dykstra, whose groundbreaking special effects achievements include the original *Star Wars* film.

Although creating a diminutive computer-generated mouse may seem like a trifling task to the average viewer, Dykstra paints a formidable picture of the work's scope. "I got brought into this picture as a supervisor, with a task almost as daunting as that of *Star Wars*," he explains. "In a sense there was no certainty as to how successful we would be in creating the character and bringing him to life. Adding fur and clothes and integrating him into live-action were all big challenges."

Throughout production of *Stuart Little*, SPI ran Alias|Wavefront Maya on Windows NT and SGI workstations as their primary app. But to meet the challenges set by the film, SPI had to assist Alias in maturing its then-fledgling cloth plug-in as well as write some of its own plug-in code for use with SPI's proprietary fur software. In regard to the challenge of creating realistic cloth and fur, Dykstra said one reason they chose Maya is that it will "support the areas that we were going to expand into," such as cloth simulators.

Dykstra's approach to creating effects starts with the audience. "Here is a creature that's within the scope of most people's experience. People know what fur looks like. They have seen animals move and know the difference between an animated character and a live animal. Verisimilitude in both those areas was critical, so it was one of the things we focused on first." When creating a character entirely from imagination, certain liberties can be taken with realism. When it comes to a mouse, however, suspending a viewer's disbelief is no small task.

Rodent Previz More than nine months of R&D went into the production of *Stuart Little* before any actual filming or final animation was done. Fur, cloth, and character design were all thoroughly tested, designed and redesigned. Like fictional royalty, Stuart was sculpted, groomed, and costumed with detailed attention paid not only by the effects crew, but by everyone involved, from studio heads, to marketing people, to the director. The mandate was that Stuart be convincing as an actor as well as extremely cute. His creators studied the essence of cuteness in depth, looking to baby animals and cartoons for inspiration.

Henry Anderson, the project's animation director, helped oversee Stuart's character design and construction and juggled cuteness with Stuart's other requirements. The modelers started with a human body and changed the proportions to create Stuart. "Stuart's proportions had to be cute and appealing, but his design had to allow for running, jumping, climbing, and dancing around," says Anderson. "If his legs had been half as short as they are now, it wouldn't have worked very well."

Modeling The design and modeling team used a fairly standard process to build Stuart. Many character studies and drawings were created along with physical maquettes. The maquettes were 3D-scanned and brought into Maya. Because the character went through such extensive design revisions and changes, most of the final modeling was done by creating and stitching together more than 200 NURBS patches.

Stuart became a subtle blend of cartoon cute and realistic rodent. Anderson says, "One of the more difficult things about Stuart is that his head is kind of cone shaped. We rounded it to make it a little more appealing, but it's basically mouse-like. His eyes are on the sides of his head, not on the front like a human's. The challenge was to make his face read with human expressions that the audience could understand." 3D sculptor and lead facial animator Kevin Hudson tried many facial permutations before settling on a final design. Variations were tried, such as adding rodent-style incisors, but it hampered Stuart's ability to speak clearly, so a more human set of teeth was used.

For Stuart's face to be able to move with the qualities of a human face, Hudson and the design team created a detailed facial structure. Instead of arbitrarily morphing from one facial expression to the next, Stuart was designed to appear as if underlying muscles control his face. Hudson says, "We tried to do the facial muscular set of a mouse, but we quickly found that there was no way to get the kind of facial expressions we needed.

ALL IMAGES COURTESY OF SONY PICTURES IMAGEWORKS.

Unlike humans, mice don't do much facial communication. We ended up taking a mouse skeleton and visualizing what human musculature would look like if it were on a mouse skeleton. We worked with an artist named Carl Nass, who is the life-drawing instructor at Sony. He has a tremendous knowledge of anatomy along with a strong artistic sense."

Making it Move While senior digital character animator Todd Pilger was able to construct Stuart with out-of-the-box Maya tools, one aspect of production required a more custom approach: The animators needed an IK solution that accommodated frequent geometry redesigns. Normally, an IK skeleton is linked directly to the geometry, where each bone has complex influences specific to a given mesh. However, Pilger says, "I didn't receive the final geometry until the last month of production. In the meantime, I had to have a way to swap in and out new versions of the geometry." Pilger solved the problem by creating a deformation lattice around the initial set of geometry provided to him. The IK skeleton was then linked to this deformation lattice instead of being linked directly to the geometry. This enabled Pilger to quickly insert new versions as he received them without having to re-create bone influences each time.

Pilger's IK skeleton was simple enough to accommodate a broad range of animating styles, but robust enough that the animators couldn't break it. Besides the normal bones and joint configuration of a typical bipedal character, joints and controls were added for curling fingers into a fist, wiggling the ears and nose, and opening and closing the jaw. Specific controls were added to the skeleton as needed on a shot-by-shot basis. This typically happened when Stuart needed to hold an object, which required a new level to be added to a hierarchy.

Small Mouse, Big Role Stuart's acting was pivotal to the movie's believability. To make the performance consistent and expressive, the team of 30 character animators, which met regularly to discuss the execution of the character's personality, comprised a broad variety of backgrounds, including 2D, 3D, stop-motion animation, and so on. While motion capture was considered early on, it was quickly decided that all of the

Stuart was rendered in several passes—one each for head, hands, clothing, and so on—then composited. That way, if one part needed to be tweaked, the whole scene didn't need to be rerendered.

character animation work would be done through keyframing.

Anderson and the rest of the team paid great attention to detail, using Disney's character animation as inspiration. For physical reference, they turned to acclaimed movie and Broadway comedian Bill Irwin. Anderson says, "We spent some time early on videotaping Bill and getting some ideas about how Stuart might move around. We came up with a walk based on some things Bill had done and added in a bit of Buster Keaton. Stuart sort of leads with his foot, swinging his toes up and landing with his heel. That helped him walk in a natural way with his large feet."

The animators blocked out each scene by positioning Stuart at key points to create a rough of the overall timing. They gradually filled in the rest of the animation, making every portion of Stuart's body as expressive as possible. "Even his whiskers were keyframed," said Anderson, illustrating the team's obsession for realism. "His ears are a very good mood indicator. If Stuart's excited, they perk up a little bit. When he's really sad or thoughtful, they droop down slightly. We animated his ears as if there are muscles on the top of his head that tighten up or relax depending on his mood."

Talking the Talk After the character animation for a scene was completed, Hudson's crew of four animators would create the lip-syncing and facial animation. To accommodate the wide range of expressions and subtle variations, nearly 60 separate morph targets were created for the head. This allowed the animators to produce an enormous variety of facial expressions by using sliders to blend the morph targets.

The morph targets included an extensive set of phonemes for accurate and convincing lip-sync, many of which also had alternate versions. Anderson explains, "For example, there are a couple different ways an M can be used in a sentence, depending on what sound is before or after it. We had to build a really broad, sort of pulled back M and one that was more pursed." Some morph targets were also created on-the-fly as the needs of the production increased.

Stuart expresses a broad range of emotion and depth of feeling. Hudson explains, "He is an adopted child who is trying to find out who his real parents are. There are feelings such as 'who am I,' and an attitude to go with that." The facial animation and lip-syncing were done to the voice performance of Michael J. Fox, who was videotaped for animation reference. The animators started with a neutral head and modified it into the desired facial expression. They looked at each transformation to make sure that the

morphing took place along lines of the fictitious musculature in Stuart's face so that all the facial expressions were logical.

Stuart's virtually black, featureless eyes posed a particular dilemma. Hudson relates, "Expressions that needed to use the direction of the eyes were difficult. They were accomplished by changing the shape of the eyelid edges around the eye."

Top: The still-dry Stuart looks out of the washing machine at Snowbell the cat. Above left: John Dykstra (left) and animation director Henry Anderson. Above right: CG supervisor Jim Berney, visual effects supervisor Jerome Chen, and CG supervisor Jay Redd.

The breadth of shots in certain scenes came into play as well. In broader shots, Stuart appears small on the screen, and even though he is little, his expression has to be clear from a greater distance. This was sometimes achieved by giving him broad, slightly exaggerated theatrical expressions. Hudson sums up the performance work by saying, "The character expressions had to be believable, charismatic, and honest, as well as cute. At no point could Stuart not look cute."

All Fur One Creating and animating realistic hair and fur continue to be one of the most technically challenging areas of 3D work. Along with ILM and DreamQuest Images' 1998 movie *Mighty Joe Young*, SPI's work on *Stuart Little* now stands as one of the best examples of digital fur to date. But it also highlights the complexity of the problem on both the technical and creative sides and dictates that, to get realistic fur, exhaustive yet equal attention be paid to modeling, lighting, rendering, and compositing.

SPI was unable to find any off-the-shelf fur tools capable of the kind of realism they needed for *Stuart Little*. So during the course of the film's lengthy preproduction, they created their own fur software. The solution consisted of modeling tools for Maya that allowed artists to "groom" Stuart in a 3D environment and custom code that plugged into Pixar's RenderMan for achieving the 500,000 individual hairs on Stuart's head and hands.

Dykstra worked extensively with visual effects supervisor Jerome Chen and CG supervisor Jay Redd to address the many aspects of fur creation. According to Dykstra, one of the key issues that needs to be addressed when creating realistic fur is the reaction of the fur as muscles move beneath the surface of the skin. "When Stuart smiles and we create a round form on his cheek, how does the fur conform to that roundness? Does it splay the way it should or does it lie flat? How does that fur relate to the normals of the surface? If you bend the surface, the fur actually spreads, like when a dog's hackles come up."

What became evident early on was that careful lighting was integral to selling the reality of the fur. Dykstra explains, "Because fur is transmissive, as well as transparent and reflective, it's got a variation in specularity depending on how close to the base of the follicle you are. These were necessary commodities in the creation of this character." He continues, "Hair is like smoke. It has volume. Many hairs together form a combination of negative and positive shapes. We had to make this hair look soft, and it had to diffuse light, refract light, and reflect light in the way a volumetric source would."

One of the team's first approaches was to give the fur a self-illumination element to simulate light bouncing among the individual hairs. In early tests, it looked like the fur was glowing, so they added a diffuse quality to it and adjusted the lighting model to take into account the diffuse color of the geometry underneath the fur. Then they factored that in to how the fur would be lit in the scene.

Chen explained the next issue the team faced. "Attention had to be paid to how a hair would receive shadows from other hairs and how it would bounce light. The hairs appeared too hard if they bounced off too much light, but the surface appeared too smooth if the shadows were too soft. Furthermore, these qualities had to be redialed for each shot, and there was no magic number." Too much specular lighting caused the fur to look oily, and not enough made it look too dry. The team had to experiment until they found the right amount of specularity to make light travel across his fur realistically.

"One of the things we discovered," said Dykstra, "is that if you don't have a highlight

on the fur that's white—and it could be anyplace—he just looks gray. A little bit of highlight someplace shows the viewer that the fur is white."

Stuart's grooming process consisted of using special tools in Maya that allowed artists to distribute 500–1,000 key hairs over Stuart's head and hands, each representing hundreds or thousands of hairs. These key hairs were manipulated to control groups of hair that would appear in the final renderings, but not in the actual 3D program. Animators had control of 27 separate information layers, most of which were controlled by separate, grayscale bitmaps mapped to the skin geometry but dictating individual qualities of the fur.

There were bitmaps for fur density, overall color, color variation, base color, tip color, twist, waviness, taper, transparency, width, and more. For example, using the density map, artists could make less fur around the nose and ears so that the underlying pink skin could show through in those areas. Once again, the team used in-depth references, this time turning to staff anatomy expert Stuart Sumida, who helped with careful studies of mouse fur and skin.

To be faithful to true mouse fur, a small amount of larger, thicker hairs were created, which also made Stuart appear furrier on film. Chen explains, "These longer hairs stick up higher than the other hairs, adding realism to the fur by catching light and helping define Stuart's silhouette."

All of the information in the key hairs and the array of defining bitmaps was passed to SPI's proprietary RenderMan plug-in, which they call their Fur Interpolator. This software called upon RenderMan's curve primitive and arrayed it across the surface of the geometry for rendering. The curve primitive consisted of flat geometry with a gradient image texture mapped to its surface to imply the roundness of a hair.

A rough version of Stuart's head, containing far fewer hairs, was used by animators to create quick test renders. It was also used in scenes where Stuart was far away from the camera. This way, his head still looked furry in shots where, normally, the denser fur of the regular head would have blurred together.

Thanks to proprietary rendering and compositing software, CG Stuart fits right in with his live-action counterparts.

Clothes Make the Mouse Almost as challenging as Stuart's digital fur was the creation and animation of the many costumes he wears during the course of the film. For this, SPI looked to Alias|Wavefront's cloth-simulation software. Starting with alpha code, they worked extensively with A|W throughout the production and contributed to many of the features that Maya Unlimited users find in the current Maya cloth plug-in.

Dykstra explains some of the reasoning behind SPI's approach to the cloth. "You're always trying to create a pipeline as flexible and as automated as possible. We had the ability to put wind into the clothes in the sense that you could say 'make the wind fast and make it come from that side' as opposed to 'vary the vertice's connective spring constant by a value of 7.' It was a much more intuitive process and pipeline because of the way they designed it."

For that added touch of realism, SPI had the costumes designed by a traditional costume designer. It became apparent that best approach was to make the patterns in the computer. The team then had a real-world tailor come in to examine the way a costume fit on the onscreen character. The tailor would instruct them on any necessary alterations, such as "bring the sleeves up a little" or "let the jacket out a bit in the shoulders." They would modify the pattern and put it back on the character in the computer environment. "Real tailoring was the true paradigm for the way we made the costumes," Dykstra said, "which was very cool."

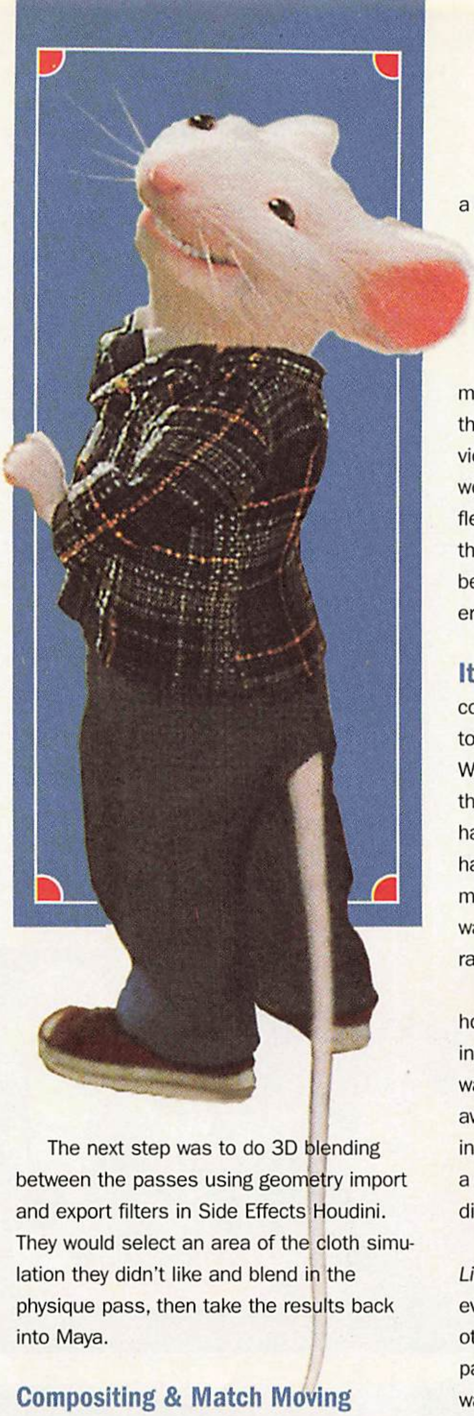
Stuart's size caused another problem for the

cloth simulation. The team set out to make the cloth look slightly stiffer than normal cloth, to maintain the sense of scale. Rather than putting Stuart in a silk shirt, they wanted it to look like a piece of silk that had been made into a tiny shirt for him. The animators found, though, that when they ran the simulation on costumes as small as Stuart's, very few wrinkles occurred due to the thickness of fabric that size. This gave the costume a quality of a doll's clothes, which was too stiff and undesirable.

The solution was to increase the scale of the cloth simulation parameters without changing the scale or actual geometry of the cloth. The simulation was run as though the costume was for a character 30 inches tall instead of six. This gave Stuart's costume visible wrinkles and draping qualities without losing the sense of his diminutive size.

Due to the heavy demands placed on the simulation software, the process was not without its difficulties. "As you start layering pieces of cloth on top of each other," Chen says "the collision detection gets pretty complicated." The team not only found humor in its failings, however, they were motivated to innovate even more. Dykstra shares an anecdote, "The interesting thing about the clothing was, when you'd get it wrong, it would just fly off of the character. There were some really funny cloth simulations where the character would go halfway through the scene and suddenly all his clothes would rip off!"

Usually, the simulation would have to be run several times, and the results from the files were blended together to make one usable simulation for the shot. Senior technical director Mike Travers tells how this was accomplished. "We used MEL, Maya's scripting language, to the hilt. When an animator would hand off a file, we wouldn't even open the Maya scene file for the first two passes of cloth. We would do it all by command line." He continued, "The Maya script would take the posed cloth [arms and legs extended in the style of da Vinci's Vitruvian Man], drop the baked-out character animation into the file that had the costume in it, put the cloth in the start position, and simulate two passes. One was a regular simulation, and the other was a physique pass, where we created mesh constraints for every point on the cloth that would stick that point to wherever it was on the body."



The next step was to do 3D blending between the passes using geometry import and export filters in Side Effects Houdini. They would select an area of the cloth simulation they didn't like and blend in the physique pass, then take the results back into Maya.

Compositing & Match Moving

Although a time investment up front, breaking the components of each scene into 2D layers gave SPI more control during the compositing stage, saving a lot of time in the long run. Stuart was rendered in separate passes for his hands, head, eyes, shirt, pants, feet, tail, and more. If color or lighting was wrong on one layer, for example, the team could adjust that layer without having to rerender the whole scene. SPI used Alias|Wavefront Composer for compositing the many 2D layers.

In keeping with a recent trend among big-budget effects studios, SPI has shied away from motion-control camera work for live-action footage. Techniques for matching the movement of a virtual camera to that of a live-action camera have improved dramatically in recent years, making the process of integrating the digital animation with the live-action footage much less cumbersome.

All of Stuart's scenes were match moved. Virtual sets were created based on the geometry of the actual sets, which provided the surfaces upon which the character would perform. Match moving allowed the flexibility for the people onstage to do things they wanted to do intuitively, as opposed to being dictated by the motion-control camera's limitations.

It's a Wrap "For me, this movie was a complete breakthrough," says Dykstra. "I got to be involved in the performance aspects. We'd design sequences, and I wasn't put in the position of having to say, 'no you can't have the mouse in the water, no you can't have the mouse dressed in this color, no the mouse can't walk on somebody's hand.' It was wonderful to become a creative collaborator as opposed to a naysayer."

The production of *Stuart Little* illustrates how filmmakers and animators are benefiting from the maturing of 3D animation software. Digital effects production is moving away from technically burdened artists laboring away at command-line interfaces, toward a field where animators are able to more directly leverage their creative skills.

What may be even more central to *Stuart Little*'s success was the SPI team's highly evident and charming quality of holding each other in high regard. In a team tightly packed with industry heavy hitters, there was certainly the potential for large egos to clash. Dykstra says, "I learned as much as I brought to this project." Each crew member gave a sense of enjoying the chance to share and learn from the others on the team. One could certainly sense the positive effect this had on the whole production. Stuart Little's adoption papers most likely had "plays well with others" on them. Maybe SPI's application forms have the same checkbox.

Barrett Fox is cofounder and president of infoplasm, a San Francisco-based web animation studio that creates *The Carpal Funnel* and the *Information Overload Overlords*, which can be seen at www.infoplasm.com.


THERE'S ONLY ONE
BIG KAHUNA!
Enter to Win the Year 2000 BIG KAHUNA Award
22 hours spent modeling
55 texture maps
4 scene files
20 sleepless nights
8000 frames!
16 renders later
1 FedEx envelope
1 BIG KAHUNA!
Now accepting entries to the 2000
Big Kahuna Awards!
Go to www.3d-design.com
for all the details.
3D
3D magazine • Part of the Miller Freeman 3D Group
Artist – Alec Syme, www.alecsyme.com

by DAVID DUBERMAN

Hair, There & Everywhere

Hair & fur tools for MAX, LightWave, Maya, Softimage & EI

Spider made with Glassworks Ltd.'s fur technology.

Giraffe made with Maya's fur plug-in.

Computer graphics folks talk a lot about textures, but in that context, it's something of a misnomer. They're usually referring to flat images applied to 3D objects to make them look more realistic. The real world is full of textures that can't easily be re-created inside the computer: cloth, stucco, tree bark, pollen on a flower, and so on. That's why virtual worlds often look flat and don't fool us for a moment.

Especially problematic is the task of representing humans and animals on the computer. Of the surfaces we see every day, perhaps the most difficult to reproduce realistically is hair or fur, whether on a person's head or a creature's pelt. Hair's difficult-to-mimic properties include translucency, flexibility, and slenderness, but what really stands out is its abundance. A typical person's head has 100,000 to 150,000 strands of hair. To model and manage that level of geometry by hand with a 3D program is beyond normal human capabilities. Even if it were practi-

cal to create the geometry, to render it would take days per frame even at the fastest render farm.

Fortunately, computer graphics programmers are good at faking complex real-world phenomena. Pixar's *Toy Story* (1995), the first 100 percent computer-generated mass-market film, used volume shaders to successfully reproduce human hair on Andy, Sid, and Hannah, the human children characters. But it wasn't easy. For instance, the shader for Andy's hair (12,384 strands) took an estimated nine months to write.

As usual, this advanced technology has trickled down from the stratospheric realms of studios such as Pixar and ILM. Gamers are familiar with the long reign of bald or helmet-headed characters in 3D games, but these days, users of most 3D apps can bestow hirsuteness upon their virtual characters. Hair/fur plug-ins come with or are available for Discreet 3D Studio MAX, NewTek LightWave, Play Electric Image, Alias|Wavefront Maya, and Avid Softimage 3D. I'll try to give you a hands-on feel for the plug-ins for the most widely used programs—MAX and LightWave—and briefly describe the rest.

(Incidentally, with the exception of FiberFactory2 and Trooper, all of the plug-ins included in this article achieve their effects with shaders, which create the appearance of fur or hair algorithmically at render time without the use of geometry. You can find technical information on fur shaders in the online SIGGRAPH proceedings library at www.acm.org/pubs/contents/proceedings/series/siggraph/. Note that this is a pay service.)

Shag: Hair The most powerful hair plug-in available for a 3D package under $10,000 is Digimation Shag: Hair ($495), which works with MAX. Developed by the same team that created its predecessor, Shag: Fur (Stanislav Evstatiev and Ivan Kolev of Dimension Design AG), Shag: Hair takes the original concept and runs with it, with a couple of exceptions. First and foremost, in the new version, you can define the hair with spline or NURBS curves. Shag: Fur provided a few parameters for modifying hair shape, but Shag: Hair gives you extensive control. Also very important is that hair can participate in dynamics simulations, such as being blown by the wind or bouncing around in response to the character's motion.

Shag: Fur was implemented as an atmospheric shader only; the new kid can also produce geometry from the hair at render time. It's not clear why this was implemented. It can provide slightly better anti-aliasing and motion blur but tends to consume massive quantities of RAM and CPU time. The atmospheric hair can cast shadows and take on other characteristics of mesh objects by means of Shag: Render, the included prerendering engine. Other improvements include support for clumping, a flat-strand option that produces realistic grass, faster-rendering thick hair, and the ability to use graphs to vary parameters such as thickness over the length of the hair strands. And as an added bonus, Digimation includes a free copy of Shag: Fur 2.0, the upgraded version.

The basic workflow is fairly simple: First, you create or load your model and define where you want hair to appear, typically by assigning a unique material ID to follicle-enabled polygons. Then you create a prototype strand of hair by drawing a spline or NURBS curve. Next, apply the Model Hair modifier to the strand and set some parameters. Like its ancestor, Shag: Hair needs special versions of the MAX lights (included with the plug-in) to render shadows from the volumetric hairs; add a couple of these, or just click a button to convert the existing lights in your scene to the Hair equivalents. Finally, you add a couple of Atmosphere effects in the Environment dialog, do a little more setup in one of these, and you're ready to render.

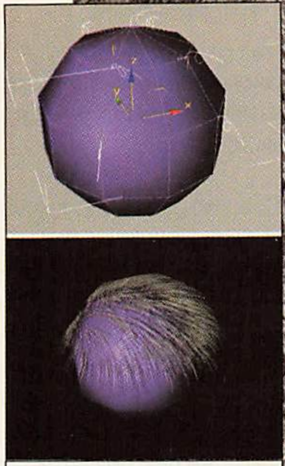


FIGURE 1.
A Shag: Hair object in 3D Studio MAX's viewport (top); the same object rendered (bottom).

Modeling Hairs Let's take a closer look at the Model Hair modifier. Its primary function is to let you determine its orientation, as well as where on the strand it connects to the emitter object. The default method, called "Mesh Vertex, Normal," lets you move or rotate the model hair to affect how the hair lies on the emitter. For best results, place the strand on the emitter, in among the other hairs. Incidentally, when you work with Shag: Hair in the viewports, the wireframe strands emanate from each vertex; the final, rendered hairs are interpolated from these. Figure 1 shows a comparison of the vertex hairs in MAX's viewport with the interpolated "middle" strands in the rendered output. The only way to increase the number of viewport hairs, to get a better idea of the final result, is to increase the emitter's mesh resolution. But this can screw up how material IDs are assigned to polys, so you need to do some careful planning from the get-go.

If you just want the hair to stick out perpendicularly to the emitter surface, the "Curve Base, Tangent" method makes it automatic. If you want maximum control over the hair's orientation during an animation, the third method, called "Vector Base, Vector," uses a special vector helper object to specify the coordinate system.

Other Model Hair modifier controls let you offset the hair from the emitter and use a graph to define the degree to which the strands are attracted to the model hair. Then there's Influence, which lets you mix multiple hair strand shapes depending on their distances from each other. You can use vectors to modify influence and even change influence over time to animate the blending of different hair strands. Of course, you can use an animated strand to begin with, and the hair will follow along. This isn't trivial stuff; there's a lot of power here and the necessary complexity to go with it.

The supplied tutorials don't really cover the upper-echelon methods, so you'll have to do a fair amount of experimentation. Believe me, it'll pay off.

The Shag: Hair Rendering Effect

As I mentioned, Shag: Hair includes two Atmosphere effects. The first, appropriately named Shag: Hair, lets you determine where and how the hair appears. Typically, you'll add one of these for each hairy object in your scene. For each, you specify the object or objects it'll affect and which parts of each emitter will bear hair: the whole thing, selected faces, a specific material ID, or a named selection set. There's also an option to affect only faces within another object's volume, although I couldn't get it to work.

It's also here that you specify one or more model hairs. You can use other objects to cut the hair, either by lopping it off or scaling it down. You can specify length, density, thickness, curliness, and clumping. Most of these let you apply a random factor and tie the setting to underlying faces' material ID. Other settings let you define how the hair grows and its color and shading, with parameters for tip and base colors plus various types of UV mapping.

Then there's the Dynamics rollout, where you can create some hairy animated verisimilitude (or surrealism, if you like), such as hair flipping around with a character's movement or being blown by the wind. Shag: Hair supports three basic types of dynamics: gravity and wind, which use space warps; and collision with other objects. This works entirely within Shag: Hair, eschewing MAX's Dynamics utility. You choose one of three methods: Simple, which is quick and dirty, but fairly realistic; Lagrange, which is more accurate; and Penalty Forces, which allows elasticity. The latter two offer more control but can be prohibitively slow, especially with

FIGURE 2. Marilyn Monroe, created by Virgin Lands ComputerAnimation using Shag: Hair. (www.virgin-lands.com)

collision. Unfortunately, collision is usually necessary so the hair doesn't go through the emitter object.

For optimal stability, you have to set the software to precalculate the dynamics simulation, with an optional keyframe-reduction step. Then you come back and render when it's done. If you want Shag: Hair to run the sim on the fly, you risk a crash. Dynamics is time consuming in any case, but especially when you're dealing with thousands of hair strands. It would be nice if the software were multithreaded, but it's not.

This second Atmosphere effect is the "set and forget" part of Shag: Hair; it's necessary simply to calculate the hair highlights, shadows, etc., and pass them to the MAX renderer. It has one rollout with settings for thick hair, memory usage, and a few others. It also lets you convert between hair-enabled and normal lights and make copies of the Model Hair modifier. Lastly, it displays last-frame statistics such as hair-rendering time, number of strands, and peak memory usage.

Shag: Hair produces strikingly realistic results, as you can see in Figure 2. (This image is part of an animation produced by Germany-based Virgin Lands ComputerAnimation, www.virgin-lands.com. The model was created with MAX.) For those who need to composite hair-rendered scenes with other images, it's helpful that Digimation includes a Video Post filter for mixing the plug-in's output into the G-buffer. Shag: Hair is a powerful, well-conceived tool for adding a whole lot of MAX to do virtual characters, whether two-legged or four, you shouldn't be without it.

FiberFactory2 Jon Tindall of MetroGrafx (www.metrografx.com) has been making LightWave 3D plug-ins for quite a while. To my knowledge, his FiberFactory2 ($200) is the only hair plug-in for NewTek's estimable 3D app. Unlike most other fur/hair plug-ins, which are based on special shaders, FF2 uses a purely geometrical approach to fiber modeling and rendering. Most of the interaction with FF2 takes place in Modeler, where you start by defining a surface (a named polygon selection) from which the fibers are to radiate. You then place the layer containing the geometry you want to add hair to in the background and bring an empty layer to the foreground, which will receive the fibers.

Next, selecting the FF2 Tool plug-in opens its dialog (Figure 3) with seven control panels. Each panel is dominated by a viewport that shows the object as a faceted solid—similar to Modeler's OpenGL Flat Shaded preview mode, but without the specular highlights. The preview, by the way, shows all the fibers, rather than a representative sampling. Still, it's quite fast, providing real-time feedback even with 10,000 hairs. That's partly because FF2 takes advantage of LightWave's unique ability to model with two-point polygons—that is, lines.

You then use the first panel (called Fiber) to select the surface from a drop-down, click Grow, and you're good to go! Well, not quite, unless you're willing to accept the default hair, which looks like your character just

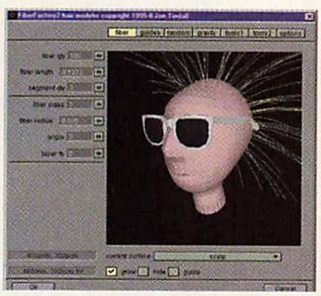


FIGURE 3. FiberFactory2 interface, showing hair with default settings.

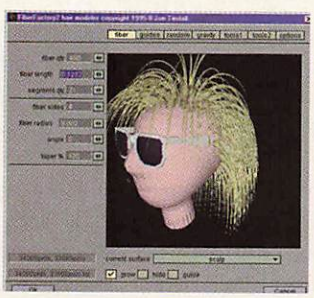


FIGURE 4. FiberFactory2-enhanced hair, with four-sided, tapered fibers.

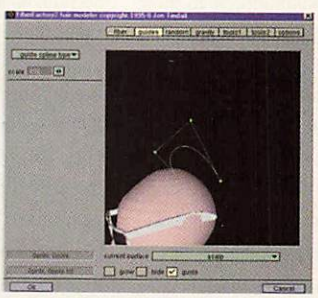


FIGURE 5. FiberFactory2 Guides let you shape the fibers with control points.

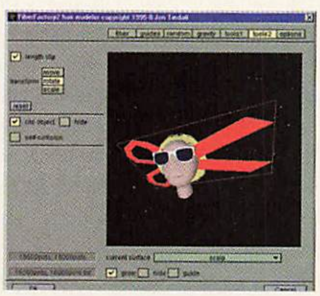


FIGURE 6. Cutting hair with FiberFactory2 clippers.

stuck its tongue in an electric socket. The Fiber panel also lets you set the number of fibers, or density, as well as the fiber length and number of segments. Increasing this last setting gives hair more of a curve, and lets it react more naturally to the Curl parameters. It's also here that you set the number of sides for the fibers. A setting greater than 1 produces true polygonal hair by extruding a polygon outline along the fiber path (see Figure 4). You can alter thickness, rotate the hair, and set the taper amount. It would be nice if you could randomize any of these, but the Random panel offers settings only for length and rotation; increasing the latter yields a messy look.

While FF2 doesn't let you use an arbitrary spline curve for the fibers, its Guides facility gives you some control by moving three control points on an approximated spline (see Figure 5) or an interpolated one. In the latter case, the curve actually passes through the control points. You can move the points only in the plane of the preview window, so you have to do a lot of rotating of the window to get the right curve. Other panels let you adjust gravity strength and direction, which determines how the fibers lie on their surface, as well as various curl characteristics, including the number of turns and kink factor.

What if the bangs are too long? Luckily, FF2 throws in a "length clip" tool that looks like a big pair of red scissors that you can move, rotate, and scale to achieve the proper "do" (Figure 6). I found this feature to be problematic; it sometimes didn't cut the hairs below its surface. This panel also includes a limited self-collision switch, which prevents the fibers from penetrating the model.

Because all these settings apply only in Modeler, you can't keyframe them for animation in Layout. However, you can use the Anchor displacement plug-in supplied with FF2 for use in Layout to add dynamics effects, like flopping around when the parent object moves. This works fairly well, but only if you specify plenty of segments for the fiber strands. Each fiber object you bring into Layout has its own surface, named Fiber_[object surface name], so you can specify different surface properties for each.

Also in Layout, the fibers can be rendered by the standard two-point polygon renderer (flat shading), shaded by a shader that renders specular and diffuse lighting effects based on the local curvature, or can be left unseen by camera. A postprocessing filter, provided by the FiberFilter plug-in, draws clusters of additional fibers based on the geometric strand. This is based on the fact that nearby hairs tend to follow their neighbors. The fibers are drawn with a depth-buffered anti-aliased line.

I found performance to be a bit flaky. The preview display disappeared regularly, and the plug-in crashed Modeler once. That could be related to the less-than-optimal drivers provided by my display card's manufacturer. While FiberFactory2 has a somewhat unfinished feel, it mostly does a fine job of creating convincing fur and hair (Figure 7), and for LightWave users, it's the only game in town.

Maya Fur Maya Fur is available only as part of Maya Unlimited—the high-end, $16,000 version of Maya 2.5, released last fall. I didn't have an opportunity to play with the software, so what follows is hearsay.

The Fur component of Maya Unlimited lets you create self-shadowing fur and short hair on multi-surface NURBS models. You can set fur attributes, such as color, width, length, baldness, opacity, scraggle, curl, and direction globally, or map them on a local basis. The procedure of adding fur to a scene goes something like this:

1. Prepare the scene. For instance, a polygonal surface must have UVs applied, and at least one light must be present.
2. Create and attach fur to models. Figure 8 shows how a furry object appears in the workspace.
3. Modify fur attributes (see previous attribute list). A powerful way to do this locally is by selectively painting or "combing" specific attributes on fur without changing the attributes of the fur description attached to the surface.
4. If you are animating the scene, animate fur attributes by keyframing the changes you make to the attributes. For example, you can make the fur appear to grow by animating its length.
5. If you are animating the scene, you

sources

Shag: Hair · list price $495
Shag: Fur · list price $295
Digimation
(800) 854-4496 · www.digimation.com

FiberFactory 2 · list price $199
MetroGrafx
(719) 748-3972 · www.metrografx.com

Trooper · list price $549
Endless Corp.
(323) 962-5656
www.endless-corp.com

Hair · charityware
Peter Watje · www.max3dstuff.com

Maya Fur · list price $15,000 (Maya 2.5)
Alias|Wavefront
(416) 362-9181 · www.aw.sgi.com

Softimage 3D · list price $12,000
Avid
(800) 949-2843 · www.softimage.com

FurShader · **Glassworks Ltd.**
www.glassworks.co.uk

Fur KIT · list price $3,021 (Green CD, including a dozen effects)
Phoenix Tools
+39 0267075747
www.phoenixtools.com

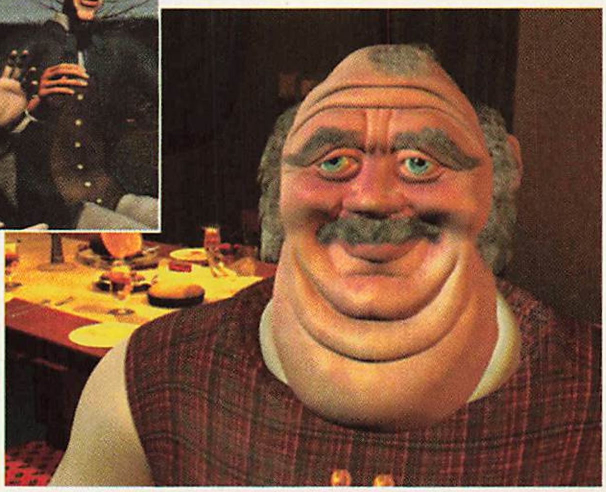
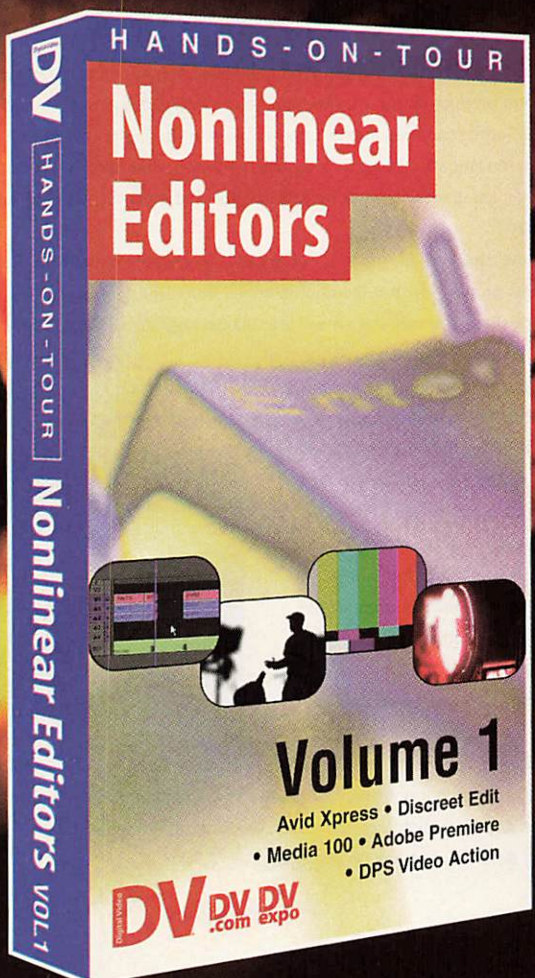


FIGURE 7. CBN Studios used FiberFactory2 to create charcacters' hair for *Night of the Headless Horseman* on Fox.

View DV's Head-to-Head Comparison of Top NLEs — Before You Buy!

Purchasing the right nonlinear editor just got a whole lot easier.



Until now, your pre-purchase testing of NLEs was limited to inspecting one system at a time. *DV*'s NLE Hands-on Tour, Volume 1 allows you to view multiple systems performing side by side, executing identical steps for the ultimate comparison. As *DV*'s Editors take these top NLEs through the crucial tasks video editors undertake every day, you'll see and experience first-hand how each of these systems manages a full project from start to finish. Supplement *DV*'s written reviews with head-to-head test-drives that reveal which system is the most elegant, the simplest, or the most intuitive to use.

Volume 1 compares Avid Xpress, Media 100, Discreet Edit, Adobe Premiere, and others. (Later volumes will feature other leading NLEs.)

Make an Educated NLE Purchase.

Order your copy today! Call 800/444-4881 ONLY **$39.99** + TAX AND SHIPPPING

4 easy ways to order! Mail or fax your completed form, or email or call in your order.
For more information, go to www.dv.com/videoseries.

CUT HERE FOR ORDERING

☐ **YES!** Send me ____ copy(ies) of the *DV* NLE Hands-on Tour Video, at $39.99 each. Please add applicable sales tax and shipping charges ($4.95 for each video shipped in USA/Canada, $10.00 for Airmail to all other countries).

☐ **Visa** ☐ **Mastercard** ☐ **American Express** ☐ **Check/Money Order Enclosed**

CREDIT CARD # ______________________ EXP DATE ______________

SIGNATURE ______________________

NAME ______________________

ADDRESS ______________________

CITY/STATE/ZIP/COUNTRY ______________________

PHONE ______________________ EMAIL ______________

Please allow 2 weeks for delivery in the U.S./Canada; 4 weeks for all other countries. Checks/money orders must be drawn on a U.S. bank in U.S. funds only.

1) **Mail** the completed order form to *DV* NLE Hands-on Tour Video, 1601 West 23rd Street, Suite 200, Lawrence, KS 66046-9905 USA;
2) **FAX** the completed order form to 785/841-2624; 3) **Email** your order to: orders@mfi.com; 4) **Call** 800/444-4881 or 785/841-1631.

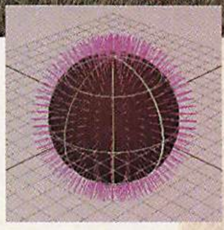


FIGURE 8 (right). Maya Fur as seen in workspace.

FIGURE 9 (above). Stripes are possible with Maya Fur.

can also add movement to fur. You do this with attractors, which are joint chains (with IK handles, linked rotations, or particles) that attract the hairs making up the fur. When you move the attractors (by keyframing, moving the object the attractors are attached to, or applying forces such as wind and gravity), the fur follows that movement according to the settings you define.

6. Set up fur shading effects. The following options are available: no shadows, no shading (less realistic); no shadows, simple shading (relatively realistic); shadows (most realistic, but longer rendering time).
7. Render the scene.
8. Refine settings and render again.

I derived this information from the online documentation, which seems very good. For instance, it contains tutorials on creating eyelashes and parting fur. There's no doubt that Maya Fur produces superb images, as shown in Figure 9. But you pay for the privilege.

Softimage Fur Two fur shaders exist for Softimage 3D. FurShader will be included with Sumatra, the next generation of Softimage, due out in early 2000. The Fur KIT, part of the Green CD ($3,021) suite of effects from Italian software developer Phoenix Tools, is available as a third-party product.

FurShader was created by engineers and animators at London-based special effects house Glassworks Ltd. (www.glassworks.co.uk), initially for in-house use. The software still isn't available commercially, but will be part of the Sumatra's Specialist Pack and will also be available as a standalone option to Sumatra users.

FurShader lets 3D artists create individual hairs on a model surface, as well as control the color, thickness, transparency, shine, and curl. It lets you design hair using IK; hairs are IK chains, integrated with the software so access is provided to all tools like selecting points, rotation, scale. Other features include self-shadowing and real-time rendering in a region (rectangular section of a workspace window). Dynamics won't be available in the first release, but should be available soon thereafter.

Phoenix Tools Fur is a collection of shaders and plug-ins for simulating fur. The plug-ins let you control the fur shape, placement, and direction on the model surface. They include PT_FurGen, PT_FurSim, and the PT_Fur volume shader.

PT_FurGen generates and controls fur over a model surface. Parameters include fur density, length, and number of segments; gravity and resistance for bending; and, in the Jitter category, turbulence, power, and curl. The plug-in outputs a polygonal mesh, with each poly representing a single hair, using the PT_FurDump plug-in to save the file in the format needed by the shader. A different file is written for each frame. It supports mesh, patch, and NURBS objects.

PT_FurSim is a dynamics simulator for fur, derived from Phoenix's cloth simulator. It performs adaptive dynamics and collision detection. Here you can set a viscosity force and a bend factor, plus parameters for stretching, friction, and mass.

The heart of the package is the PT_Fur

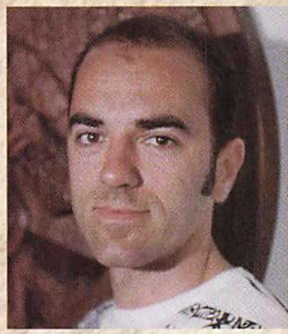
How ILM Meets the Fur Challenge

by Christian Rouet, Director of Software Development, ILM

Rendering fur is not a trivial matter. When all aspects are considered, there really isn't one product on the market that solves every problem related to the process. Some renderers support certain ways of faking fur rendering more or less realistically (using, for example, geometry and mapping tricks), but these products generally work for still images and don't address the animation of fur.

Solving the problem of rendering fur doesn't necessarily mean that the hair will move properly or that the character will be designed properly in terms of modeling. You can use an existing 3D package to try to do some of that, but it's usually very specific to the tool you use. There really is no standard or integrated solution for creating realistic hair and fur.

Most modeling, animation, and rendering tools are designed to create, deform, and render flat surfaces. Creating materials such as fur, liquid, or smoke is much more complex. It often involves combining several exotic approaches to come up with a solution that works.



To achieve a believable appearance, hair must contain subtle organic variations in density, length, orientation, and so on. These aspects must be controlled interactively because, as the creature moves, the hair is also animated. Much of it can be automated, but not every situation can be handled that way. For instance, the creature might scratch its fur, or something might fall into the hair, causing different and unique reactions.

Challenges Many different aspects must be defined for each hair and how it interacts with its neighbors. The way it moves, and especially the way the light interacts with it, is very different from flat-surface elements. Hair is also very thin, and there are many ways it can clump—cleanly washed, dirty, curly, thick, thin, dense, or sparse. It becomes impractical to try to approximate such a volume filled with complex individual hairs. You have to come up with models to approximate that.

If you look at an animal closely, you'll notice that even short fur can be very, very complex. Light penetrates fur through the thickness of each hair. If each individual hair on your model hasn't been lit properly, it will really stand out in your final animation.

When you apply a basic shadow on a rendering of a surface edge, it will look flat no matter what approach you use. The lighting, shadows, and specular highlights on individual strands won't be accurate when you group them together. There's no way it can create an effect like a real-world horse that's sweaty from running, where each hair is very specular and appears wet.

The accumulated effect of lighting hair that's been clumped together in certain directions has a very specific look to it, almost like brushed metal. This effect is called *anisotropic*, which means that the way you perceive the surface depends on its orientation to the light, unlike plaster or any other diffused surface. As you spin a compact disc, for example, it reflects light differently depending on its orientation. That's what fur does, like wet fur on a horse. This is very hard to simulate because, if you get closer, you will see each individual strand clumped into multiple hairs that form a single shape. The lighting on it will be very complex.

Proprietary Fur At ILM, we've developed stand-alone applications internally to do character animation and modeling, which were initially designed to do only skinned characters. We had to extend them to support modeling and animation of fur primitives. The fur primitive is unlike most surface elements. It has a completely different geometry. We rewrote a specific custom renderer just for dealing with hair, fur, and feathers.

A single creature potentially has millions of hairs, and there's no way to control them explicitly one by one. We use a concept called "control hairs," where a control vertex defines the surface locally. We can provide interactive control over hair on a set or subset of chosen parts of the creature. We have a process of procedurally growing nice-looking hair from these spot controls.

We provide tools so our artists can define wherever they want detail and control, whether it's the shape, length, or orientation of the fur, for both modeling and animation. This includes physical simulations such as wind, gravity, or the reaction of a foot stepping on the floor—physical simulations of what the mass is doing and how the hair should react to it. A good example of this is in *Mighty Joe Young*. When he's running, you see all the hair reacting to the run. Most of the hair—not all of it, but most—is automated through very complex physical simulation techniques, including gravity, wind, displacement velocity, and the actual run. On the rendering end, we use a stand-alone renderer that handles fur and hair specifically. Its design has nothing to do with any kind of traditional rendering system like RenderMan or Mental Ray.

The output of that rendering engine is composited in 3D to a more conventional renderer because there are complex interactions between the character's hair and other objects that might be in the way. Imagine a monkey wearing a watch. The watch isn't furry, so you might want to render it with a traditional renderer. But the hair surrounds that watch may be partially occluded, and the light will cast a shadow of the hair on the watch and vice versa. All of these elements should be rendered separately. It's far more complex than just a simple 2D comp, as it involves assembling several partial renderings. It's a very tricky integration.

All Fur One Modeling fur is still a very young field that hasn't been developed much in commercially available tools. Off-the-shelf solutions have only reached a fraction of their potential, but we're moving in the direction of more advanced tools for fur and hair creation.

volume shader for rendering medium-length fur generated by the plug-ins. It lets you specify global fur parameters such as root and tip thickness, transparency, illumination method, self shadowing, and texture maps for controlling the fur's length, density, and colors. The hair color can be global, derived from the object out of which the fur grows, or derived from any texture map. The shader supports self and cast shadows, reflections, and, being a volume shader, multiple instances and network rendering.

Endless Trooper Lastly, we have Trooper ($549), from Endless Corp. (www.endless-corp.com), which works only with Power Mac systems running Play Electric Image Animation System (EIAS). Like FiberFactory2, it's a geometry-based system for creating hair, fur, and other effects.

You define any group, such as a strand of fur, as the "prototype group" for a surface. The prototype group is then replicated over the surface of the base group. You can distribute different prototype groups evenly or nonuniformly across a surface and interpolate densities between edges and surfaces. And you can rotate and scale prototypes, with a nonuniform option for both.

Strand colors can be derived from the base object. For animation, you can morph between different prototypes and animate or deform the prototype, which affects all the copies. Trooper includes dynamics: The hair can respond to forces such as gravity and wind, as well as movement of the base object, with collision detection. For those using EIAS to produce character animation, or even wind-blown fields of grain, this could be the tool you're looking for.

Your Hair Plugs Are Showing A wide range of hair/fur software is available at a wide range of prices. But with the exception of Softimage, users of a given program don't have much choice, and in most cases, you've got to spend money before you can try the software out. One notable exception: Peter Watje's free Hair plug-in for MAX. This estimable 3D programmer operates under the "charityware" principle: If you like the software, donate money to your favorite cause. Since it's free, I'll let you discover its features for yourself; you'll find Hair (alpha version) and plenty of other free MAX plug-ins at www.max3dstuff.com.

David Duberman is a technical writer specializing in 3D graphics and animation, based in the San Francisco Bay Area. Contact him at duberman@dnai.com.

2ND ANNUAL GDC INDEPENDENT GAMES FESTIVAL
Games without a publisher are given to a jury of industry experts who review, then vote for the best games in these categories:
CATEGORIES
Grand Prize (BEST OVERALL GAME) • Art • Audio • Game Design • Programming
The Audience Award is chosen by the attendees of the Game Developers Conference, March 8–12, 2000 at the San Jose Convention Center.
The finalist games are demonstrated at the IGF Pavilion on the Expo Floor, where they'll be played by attendees, fellow developers, press and publisher representatives. The winning games will be honored at a formal ceremony at the GDC.
CO-SPONSORS
Miller Freeman Game Group AND Microsoft
INFORMATION
http://www.indiegames.com.

BY DARNELL WILLIAMS

For the Animator Who Wants It All

Nichimen Mirai 1.0

As president of an animation facility specializing in character animation, I've used quite a few 3D packages over the years. Currently, we're using our in-house optical motion capture system as a major component of our animation process. Riding the bleeding edge of technology, our production pipeline is always in development. I want perfect IK, the fastest modeler, and the best renderer—basically, I want it all. In this respect, Nichimen Mirai 1.0 offers a unique work environment.

Mirai is a complete rework of Nichimen's N-World software. N-World began in the mid-80s at Symbolics, a defunct software vendor that created the famous animated short *Stanley and Stella*. Nichimen acquired the software division and directed its tools toward the game market. N-World gained a strong following among game developers, primarily as a development tool for the Sony PlayStation and Nintendo consoles.

Nichimen's development staff has taken the strongest aspects of N-World—most notably the polygonal modeler and animation scripter—and improved them through a complete software redesign. From its windowing technology to interface philosophy, Mirai offers a unique and interesting approach in what is already a very competitive market. Mirai's particular strengths position it primarily as a system for character animation.

Running under Windows NT and IRIX (plus a soon-to-be-announced Linux version), Mirai's performance is impressive. I tested it on both an SGI Octane and a 450MHz Pentium II Dell with a Diamond FireGL 3000. I opened four geometry editor windows displaying several 20,000-plus-polygon textured objects, a 3D paint editor, and two animation viewers, and the interaction was still very responsive.

System requirements for an NT platform include a 250MHz Pentium II, 300MB of RAM, 400MB of virtual memory, an Ethernet card with TCP/IP networking installed, and an OpenGL graphics card. (For a list of compatible cards, see Nichimen's web site, www.nichimen.com). A three-button mouse is necessary to run Mirai, and it uses all three buttons very efficiently. SGI systems must have IRIX 6.3 or higher, an r4400 or better CPU, 128MB of RAM, 300MB hard disk space with at least 300MB of swap space, and 2GB of virtual swap.

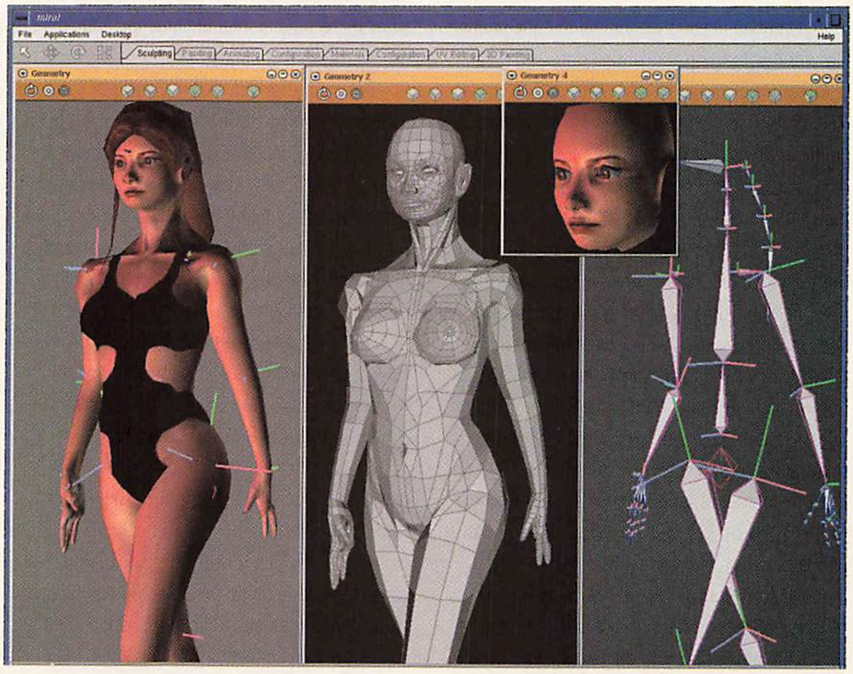


FIGURE 1. This window shows four unique geometry editors. Each displays a different aspect of the same 3,000-polygon character. Viewer 1 shows the textured object, Viewer 2 displays the same character with textures disabled, Viewer 3 displays the IK skeletal structure, and Viewer 4 displays only the face part for detailed editing.

Overview Mirai starts to differentiate itself from competitors in its approach toward 3D content creation. Where other software packages allow for a single modeling or animation window, Mirai offers editor windows. Editor is Mirai's term for an application window opened in the workspace. By working across multiple editors (Figure 1), you can use all Mirai's tools—modeling, animation timeline, 3D paint, UV editing, etc.—to edit the same data, rather than the more traditional single-element system used by most 3D apps.

With the editor system, I found that the hindrance of switching modes to accomplish a given task is gone. By creating a layout of any number and type of editors, you can modify an element at any time and any place in the workflow. You're able to edit the geometry of an object while modifying its skeletal animation, for example. Or, as you touch up the model's textures in 3D paint, you see the changes in real time as the synchronized editors update automatically. This is even possible while working on several different characters simultaneously.

Mirai file structures are based on scene files rather than a file for each object or material. With the option of loading subscenes, several animators can work on a master scene while focusing on their particular duties in the production and see their changes reflected correctly. This is a nice idea in theory, but in practice I found the problem to be that all scene elements (such as objects, maps, and materials) are written to a single directory rather than to a structured set of subdirectories. It becomes confusing when you need to manage files out-

side the Mirai environment. If you want to load a single object from a scene file containing 1,000 objects, you must load the whole scene file. Depending on the size of the scene file, this could prove a real time waster. Working on separate, simple scenes is a reasonable approach, but I still prefer the option of being able to load elements from a scene file without loading the entire dataset.

Clean User Interface The user interface is clean and well laid out. This made learning Mirai fairly quick, even pleasant. The interface avoids the clutter of too many icons and offers you a simple, almost minimalist work area on startup. There are six application-specific tabs that run along the top of the screen and create the primary editors: Geometry, Animation, Paint, 3D Paint, Materials, and UV Editing. Each editor has its own set of sub-editors that are used only when applicable to your current task.

The interface has several stylish features, such as context-sensitive menus. As you pass your mouse over an editor, it's instantly activated. There's an automatic tiling of windows when you build your workspace layouts, and you can save your custom configurations for use in other projects. Editor-specific hot keys and inventive use of the three-button mouse are great productivity enhancements. The mouse buttons follow a set paradigm, regardless of which editor you're working in. The left button makes selections, the right button edits selections, and the middle button is for the camera. The click options are editor-specific, but the changes still reflect a logical order. A minor, but irksome point: The input sliders used throughout the editors aren't as sensitive as they could be, which makes them difficult to use. I couldn't easily scroll to a selected value in an animation channel using the slider. Typing values is more efficient, but I wish there was some form of sensitivity control to make the sliders faster or slower.

The scene graph editor provides general management of the current scene. All the contents are displayed: geometry, images, scripts, materials, and skeletons. Hierarchical structures of objects and scripts are visible, as are the associations between different elements, like how a particle property or material relates to a polyhedron, for example. Several scene editors can be opened to manage several loaded subscenes. Elements can also be dragged and dropped between entirely different scenes.

Making Geometry The geometry editor uses a winged-edged polygonal-modeling paradigm, which describes an object as a closed surface. A volume technique lets you maintain a true sculptural approach. The modeled form can be changed much like clay. Nichimen follows the philosophy that polygons are the best way to model, thus avoiding all the inherent editing problems of spline-based patch surfaces. Typically starting from a primitive object, you can edit any element—vertices, segments, faces, and edges—and create some startlingly complex organic models. I found it to be the most tactile modeler I've worked with.

The subdivision surfaces feature lets you use low-resolution control models that drive smoothed, hi-res meshes (Figure 2). Unfortunately, any modification made to the control object is automatically reflected in the smoothed hi-res mesh. Through the geometry editor menu and element-sensitive bar, you control object editing, camera manipulation, object display, restructuring, renaming, and rendering. A powerful feature is virtual mirroring, which allows you to interactively model half of a character and see the changes automatically on the other half. With the selection of a group of faces, a face part can be named and recorded into the model. Select all the faces that were used to create a nose, for example, and save them as a part. Then the nose part can be viewed and edited separately from the main object.

In addition to its plethora of polygonal editing tools, the geometry editor can record multiple displacement states or morph targets on the same body (Figure 3). Displacements are recorded positional differences of a model's vertices relative to a specified initial state. In facial animation, for example, you can create a wide range of expressive gestures by mixing a few displacements. Better still, you're able to make topological changes to the geometry of your object, such as adding more detail to a character's lips, without destroying any of the previously saved displacements. The new vertices are automatically updated into all saved base states along with displacements of your model.

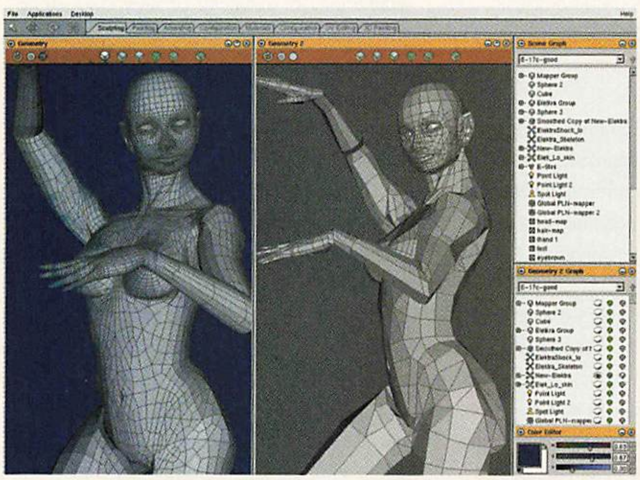


FIGURE 2. A smoothed hi-res polygonal mesh and its original control mesh. The low-res model is animated then replaced by the hi-res model for rendering. Again, this is a single-surface character.

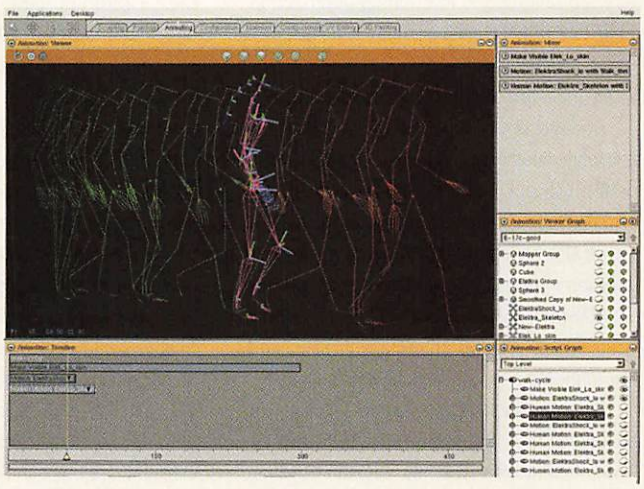


FIGURE 3. A motion capture file was read into the animation editor, and the channels are displayed in the timeline. "Ghosting" of the skeleton shows its motion over a specified range of frames.

Animation Action Animation uses several editors: the animation viewer, the timeline, the mixer, and the scene graph. Multiple animation scripts can be edited in the same scene, rather than the more common single-script approach. Scripts can also reference other animation scripts, combining them in a third unique script.

Mirai's keyframing approach is straightforward. After selecting the object and moving or displacing it in the animation editor, animation channels with keyframe values are automatically added to the current script with a simple click. An auto-keyframe option can also be enabled, in which you proceed to the next desired frame in the timeline, and an object's modifications are recorded automatically.

The display of channel data in the timeline editor is clear and functional. The top-to-bottom ordering of animation operations and the ability to reorder and deactivate an operation's priority makes it simple to change any aspect of animation. Multiple copies of the same channels can be cut, pasted, and layered, offering endless opportunities to test the output. Some of the iconography in the timeline editor is a bit obtuse. The icons are cryptic, small, and generally take some getting used to. For example, to make a channel visible, you open and close a winking eye icon in the animation graph editor.

Even though you can create nested channel operations in an animation script, there's no inheritance from parent to child of common elements. For example, to animate a bird, first I'd create a channel called Make Bird Visible. Then I'd add several subchannels to the first, such as Displace Bird and Fly Bird Along Trajectory. The Bird object should be inherited, but you must declare the animation object in each instance. The animation playback feature even creates QuickTime movies you can load into Adobe Premiere. Oddly, there isn't any audio capability. It is currently scheduled for addition into the environment "in the near future" as a third-party plug-in.

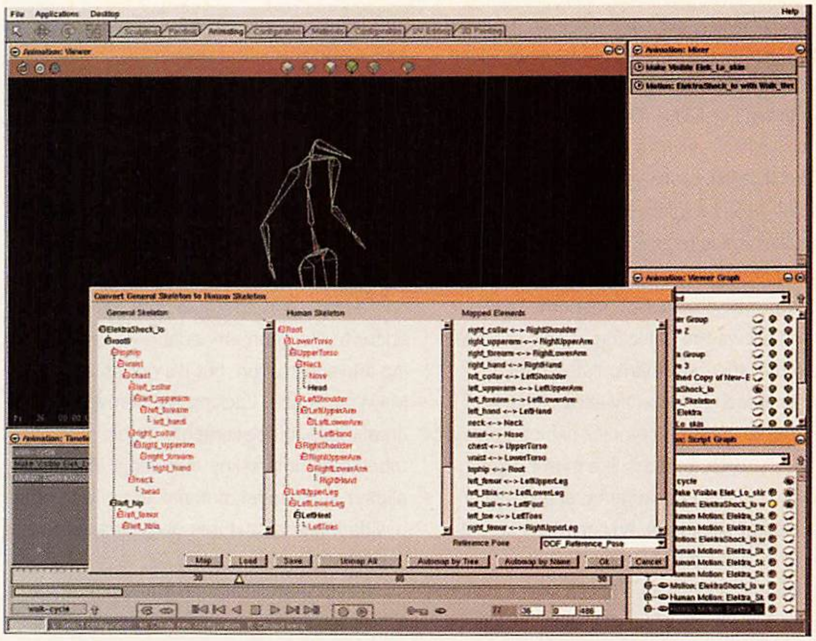


FIGURE 4. Motion capture data from different sources can be remapped to a biomechanical human skeleton using the Mapping menu. Motion blending, cycling, and skeletal transformations can be created easily using the human skeleton features.

Paint Power Paint follows a path established by Adobe Photoshop. A wide range of tools is available and can be applied with an even wider range of image processes or inks. The brushes range from texture airbrushes to image-warp tools. The inks, which can be applied to pressure-sensitive brushes and shapes, for example, include Smear, Vapor, Reveal, Blur, Expand, Noise, Scatter, Mosaic, RGBA Swap, Shift, Flip, Mirror, Magnify, Edge Key, and Chroma Key. The paint tools respond well with a Wacom tablet, allowing for a natural feel. Many of the brush shapes and inks can be animated through the timeline editor.

All the power of the paint editor is carried directly to the 3D paint canvas. Multiple map types can be specified as the targeted paint image. You can paint directly on various texture and reflection maps, including diffuse, specular, opacity, bump, ambient gel, directional gel, metal, Fresnel, reflection, and refraction maps. The speed of paint on the object surface is very nice. You also have the option of waiting to update the maps until after the camera is moved.

All of Mirai's Photoshop-style paint tools can be animated with the animation editor. I created an animation script that loaded a range of frame-grabbed images, ran several image-processing routines, composited several layers, and saved the file as a texture map. Mirai works quite well as a 2D compositor, and it's useful to have the ability to animate not only a shape or brushes path, but to also alter its opacity, color, and so on.

Super Skeletal Animation One of the best features of Mirai is its skeletal animation. It uses both FK and IK techniques for posing and incorporates biomechanical motion-editing tools from third-party developer TestaRossa. In this area, I think Mirai has surpassed Alias|Wavefront Maya, Softimage 3D, and Discreet 3D Studio MAX.

In Mirai, there are two types of skeletons: human and general. The human skeleton is a bipedal biomechanical model that allows the mapping of humanoid motion capture data—regardless of file format, skeletal proportion, or skeletal topology—to another unique skeleton. There are specific topological requirements that need to be followed when using the human skeleton with the motion-editing tools. The general skeleton allows the use of any skeletal topology, from quadruped to arachnid. When building custom skeletons, they need to be drawn first as a wire element, then converted into a

sources

Mirai 1.0 • list price $6,495
Nichimen Graphics Inc.
(310) 577-0500 • www.nichimen.com

System Requirements:
•Windows NT: NT 4.0, Service Pack 3, Pentium II 266MHz or faster, 128MB RAM; 300MB hard disk space; 300MB virtual memory
•SGI IRIX: IRIX 6.3 or greater; 128MB RAM; 300MB hard disk space; 300MB virtual memory
• Either OS: compatible OpenGL card (see list on Nichimen's web site); Ethernet card; three-button mouse.

skeleton. This doesn't strike me as the most intuitive approach and should be redesigned to incorporate drawing the skeletal chain, as in Softimage or Maya. One less step is my motto.

The IK system allows movement of selected skeletal joints in 3D space. The skeleton mimics natural movement by affecting the joints along the bone's chain according to specific degrees of freedom. The FK method allows the selection of one or more bones that the user manipulates directly. Each approach is instantly accessible, and I was able to create a complex skeletal animation easily. With features like mirror posing, tack and glue pin constraints, and physical dynamics of skeletons, any animator would revel in its powerful simplicity.

Skeletal topology can be added or deleted without destroying an applied animation. Skinning is intuitive and fast, plus there's useful new deformation lattices, soft parts, and displacement linked to joints. Physical properties such as rigid- and soft-body dynamics can be applied to the body of the model. With these techniques, you can mimic fat, muscle, and the like to really enhance character animation.

Mo' Better Capture The motion capture editor gives character animation in Mirai an even more versatile workflow (Figure 4). Motion capture data files are read as general motion files, but they can be converted to human motion and edited using the TestaRossa biomechanical tools. Some of the operations available are transitional blending, looping, scaling skeletal proportions, stretching and compressing data, modifying timing, layered transforms, mirroring, and layered key posing to motion files. You can also convert motion capture files to keyframe channels or decomposed rotational channels. The system can record and export motion data in several industry-standard formats but lacks support for real-time motion-data input, like that produced by a magnetic capture system or encoders.

Map It, Render It Mirai's attribute editor works across several properties: surface, surface material, particle, particle material, and lights in Mirai's native render domain or OpenGL. There are both raycasting and raytracing render features that perform at a competitive speed and quality. You can apply and adjust dozens of layers of maps with various blend types and unique UV coordinates, which can lead to some sophisticated mapping. Volume rendering and depth of field are also available, with motion blur and lens flare scheduled for implementation soon. Support of shaders for Pixar RenderMan and Mental Images Mental Ray is in development.

The workflow—dragging materials from one editor to another, like from the scene editor to the geometry editor—is an interesting implementation, but it's not as robust as Maya's solution. Lacking are items like procedural maps, displacements maps, and distributed command-line rendering. Mirai supplies a substantial material library that offers a wide range of textures and surfaces to experiment with.

The UV editor offers a simple interface for UV modification. Your 3D model's surface is unwrapped and displayed as an alpha channel against the image map. The object's mesh representing the UVs is displayed over its associated image map. You can select and edit the displayed alpha mesh's vertices, segments, or faces to make the desired adjustments.

Physical Dynamics The physical simulation tools control behaviors for particle systems, soft- and rigid-body properties, and ropes. Available forces include gravity fields, point fields and flow, directional fields and flow, turbulence fields, axis flows, path flows, and wire vortex flows. Particles can be rendered as points, lines, and/or re-instanced objects. Multiple obstacles can be associated to every particle group.

Particle creation follows a simple strategy: Any group of vertices, segments, and faces, either of an object or one of its named collected parts, can be assigned as an emitter. The same object and named part can also be assigned physical properties. You could, for example, apply a displacement animation to an object and have it emit particles from its twisting form, as the object itself uses an applied rigid-body property so it tumbles and bounces across a plane. All the while, the emitted particles recognize the actual surface of the object and ground plane as obstacles. Beyond obstacles, Mirai's particle system includes action operations. When a particle strikes a surface, it can itself become an emitter. In behavioral properties, a particle system becomes a "flock" and is affected by parameters such as clustering, target seeking, and obstacle avoidance, which uses visual range as a cue.

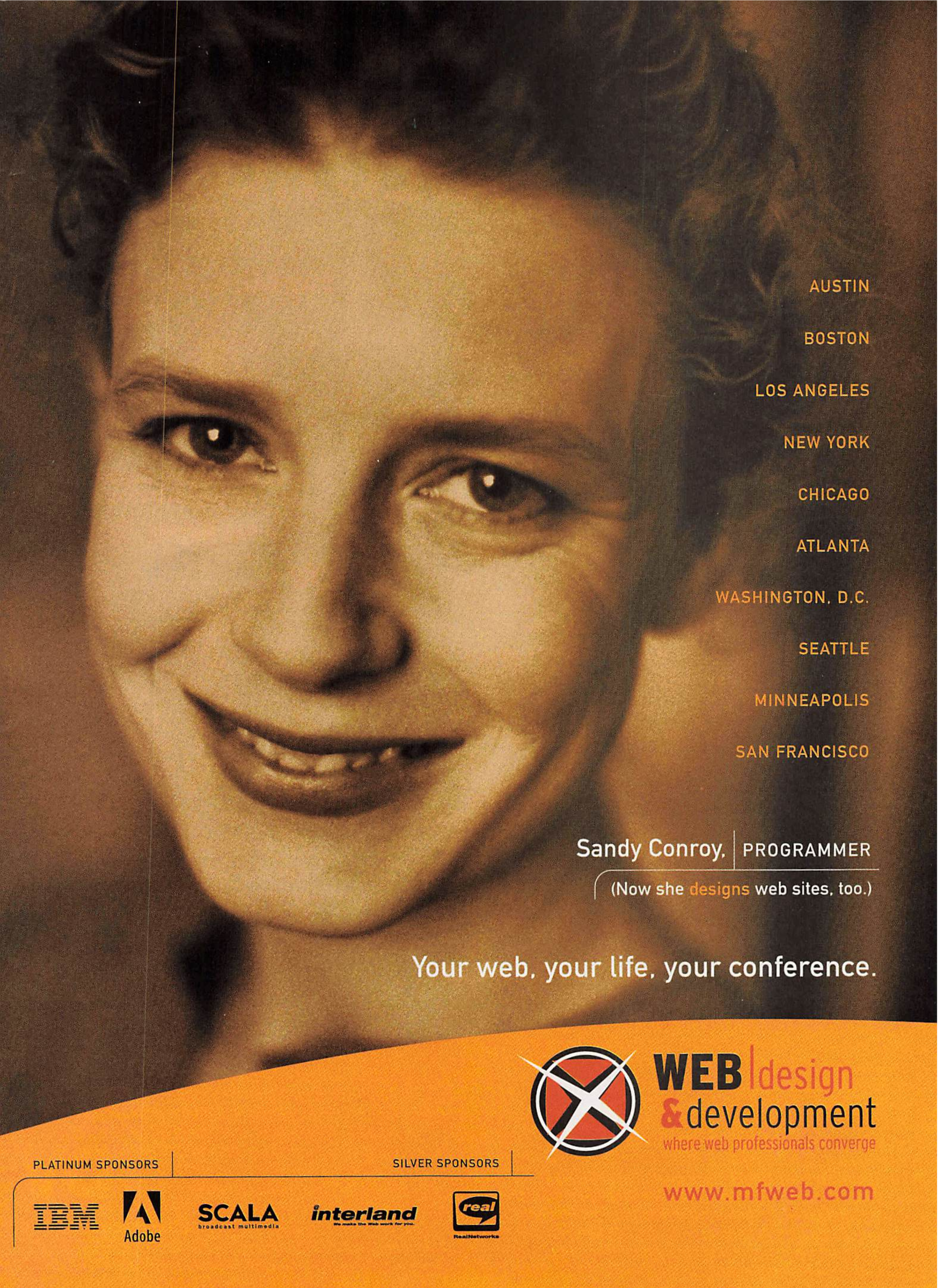
Game Exchange The tools Nichimen developed for the game market are still a major part of Mirai's environment. The color-reduction tools let you reduce the bit depth of an image, edit the associated palette of an image, and map multiple images to a color map. Game Exchange is Mirai's extensive export program. It exports a complete scene description: geometry, animation, skeletons, etc. With the popularity of next-generation game consoles (Sony PlayStation 2, Sega Dreamcast) and powerful game cards in PCs, Mirai should maintain a strong presence among game developers.

Prosperous Program The manuals, tutorials, and online help are complete and well written. I was able to find answers easily for all my questions regarding the software. The support staff also proved knowledgeable and helpful and returned calls promptly. Nichimen's web site is a well-designed and useful source of information and software updates.

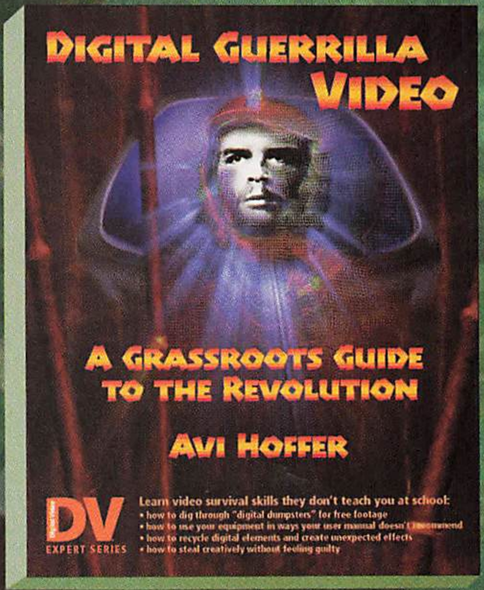
Mirai's environment is a definite improvement over its previous N-World incarnation. The new interface and editor workflow create an environment that's easy to learn and fun to work in. The design process is seamless, allowing an artist to work on any aspect of the project at any time. You can bounce from modeling to 3D paint to animating an object's physical properties at any point. If your interest is in character animation, Mirai provides some of the most powerful tools available and the best skeletal system I've seen. As an effects package, the particle system has great potential, but the renderer and attribute interface need more work.

Mirai's completeness and extensibility make it a promising candidate to be the foundation of any animation production pipeline, especially if you're creating characters.

Darnell Williams runs ElektraShock, an animation facility specializing in motion capture (www.elektrashock.com). He's been in computer animation for 14 years, with Symbolics, TDI, and Softimage. He also worked on Aladdin, Disney's first VR theme ride, and, as director of mocap at Digital Domain, L.A.'s first optical motion capture studio.


AUSTIN
BOSTON
LOS ANGELES
NEW YORK
CHICAGO
ATLANTA
WASHINGTON, D.C.
SEATTLE
MINNEAPOLIS
SAN FRANCISCO
Sandy Conroy, PROGRAMMER
(Now she designs web sites, too.)
Your web, your life, your conference.
WEB design & development
where web professionals converge
www.mfweb.com
PLATINUM SPONSORS
SILVER SPONSORS
IBM
Adobe
SCALA
interland
real

ARM YOURSELF for the REVOLUTION



Paperback with CD-ROM,
ISBN 0-87930-575-4,
$39.95

"The philosophy of the digital guerrilla: No time, no budget, elements of varying quality, limited equipment, and quick wits."

"Many books and journals have used the word 'pioneer' to describe the early adopters of digital video technology. I seldom felt like a pioneer; mostly I felt like a poorly trained soldier trying to fight in thick mud with defective weapons."

"I often fantasized about a book, a kind of manifesto, that would help me cut some corners and make the binary life a bit less painful; a book that would tell the truth about this new technology—the good, the bad, and the downright nasty. There was no such book..."

Low-cost digital tools have created a revolution in video production. In *Digital Guerrilla Video,* Avi Hoffer presents the creative "manifesto" for today's independent video producer based on the guerrilla tactics of spending little money, using few tools, and applying new rules. Step by step the author helps the reader establish a production strategy and successfully employ these guerrilla tactics to create compelling video, interactive multimedia, and Web-based projects.

Available at book and music stores everywhere. Or call (800) 848-5594.

Miller Freeman Books • 6600 Silacci Way • Gilroy, CA 95020 USA
fax: (408) 848-5784 • **e-mail:** mfi@rushorder.com • **web:** www.books.mfi.com

Distributed to bookstores by Publishers Group West.

DV

Miller Freeman Books

DV2000

BY DAVID DUBERMAN

Real-Time 3D Authoring Tools for Windows

NeMo Creation 1.0

Virtus OpenSpace 3D Author 1.0

Many 3D artists are content to produce still images, and many animators want only to see their animations play back on TV or a movie screen. But what if you want to produce an interactive real-time experience, such as a game or an interactive architectural fly-through? If you're not a programmer, you need to use an interactive authoring tool.

The market for interactive authoring tools is dominated by Macromedia Director, but Director can't do 3D (despite several plug-ins that purportedly try to provide this functionality). Fortunately, two new Windows tools can: NeMo Creation 1.0 ($990) and Virtus OpenSpace 3D Author 1.0 ($395). NeMo is aimed at the game development community, and Virtus is best suited to the low end of the interactivity spectrum, such as fly-throughs.

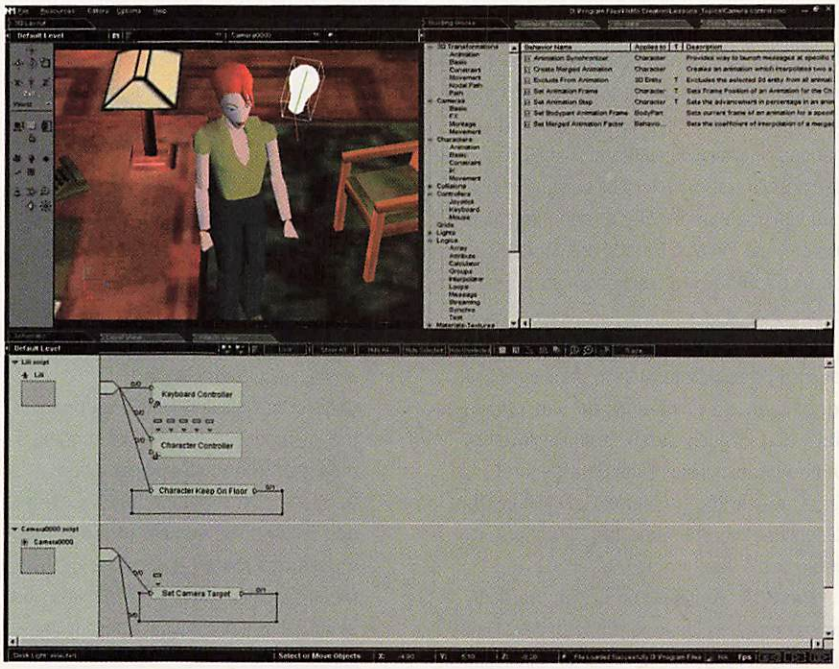


FIGURE 1. NeMo Creation's user interface provides a 3D view for scene setup and a Schematic view for visual programming.

NeMo In the old days, computer game characters lived in Flatland. You could wrangle them up, down, and sideways, but almost never in or out. Consequently, creating games was relatively simple, and many avid gamers jumped the fence and became developers without having to strain too many brain cells. But things change. Nowadays, most computer/video games take place in virtual worlds that are a lot closer to our own than were the vertical scrolling fields of yore. Development is complicated by object-oriented programming, 3D engines, physics simulation, force-feedback input devices, character animation, multiplayer functionality, and many more such issues. You still don't need to be a rocket scientist to develop games, but it's no longer child's play.

Nevertheless, the temptation to create your own games is still great among those who've tired of simply playing. With greater obstacles, it's an ideal situation in which enterprising toolmakers can produce development systems enabling the creation of relatively sophisticated games without the need to stare, bleary-eyed, at countless lines of cryptic source code deep into the night, every night. That's the goal of NeMo SA (formerly Virtools), the French company that has released NeMo Creation 1.0, a package targeted at independent multimedia authors and those who want to integrate real-time interactive 3D into their content. It lets you import 3D objects, 2D images, and audio, and add interactivity by applying behaviors to the objects.

NeMo Creation includes the NeMo core engine (the foundation of the NeMo technology), the NeMo app, a library of about 200 behavior building blocks, and the NeMo Player (as a standalone .exe file and Internet Explorer/Netscape Navigator plug-in for web deployment).

Also available from Nemo's web site (www.nemo.com) is a free plug-in for exporting Discreet 3D Studio MAX objects, scenes, and animations to the NeMo format. For high-end 3D game developers, the company makes NeMo Dev, which adds the NeMo SDK to NeMo Creation, giving developers low-level access to NeMo technology.

Creating with NeMo To illustrate how NeMo Creation works, I'll take you through a typical workflow. (Figure 1 shows NeMo's UI.) First, create your assets. The program supports three 3D file formats: .3ds; NeMo's proprietary .nmo format, which you can export from MAX and Softimage with a plug-in; and .X (Direct3D). For texture maps, you can use image files in these formats: .bmp, .jpg, .tga, .dib, and .pcx; for moving textures, .avi. Sound support is limited to the .wav format, MIDI files (.mid), and audio CD tracks.

Next, use the menu command Create New Data Resource to automatically build a directory structure, with subdirectories named Characters, Sounds, Textures, and so on. Jump over to Windows Explorer and copy your assets into the appropriate locations, then go back into NeMo to open the data resource and start building your project. Your resources appear in a tabbed window that displays an Explorer-style view, where you can see the contents of one directory at a time. Drag your assets and behaviors over to the 3D Layout, an optionally hardware-accelerated window

(OpenGL or Direct3D) that displays your scene and lets you manipulate objects.

Here's a simple example: Say you've built a hierarchical character named Davey in MAX. You'd use NeMo's MAX plug-in to export Davey along with a basic "Stand and Wait" animation, where he might look around and rock on his heels. Incidentally, the exporter can handle bipeds from Character Studio, and can optionally bundle texture maps in the .nmo file. It can export transform-based animations of entire objects and hierarchical objects, but not modifiers or sub-object animations. You can transfer a morphing animation created in MAX, but it's not very straightforward; you have to save each morph target as a separate object, then set it up in NeMo.

Once you've exported Davey's basic data, you can go back into MAX to create a number of alternate animations for him, such as Walk, Run, Crouch, Jump, Walk Backwards, and so on, exporting each as a separate file. Find Davey in the Resource view (Characters folder) and drag him into the 3D Layout, where you'll see him right away.

The 3D Layout's accompanying vertical toolbar provides icons for transforming Davey and navigating the view. A certain MAX influence is evident here, not only from the look and feel of the tools, but from the fact that you can, for instance, lock the current selection or cancel an operation by right-clicking. Alas, you can't fit/jump the view to the scene or current selection, nor is there a standard undo function, although you can set an "initial condition" for each object's position, rotation, and scale, and return to it at any time.

If Davey's in the dark, you can use creation tools to add light sources: point, spot, and two types of directional lights. Other creation tools let you add cameras, grids, curves for assigning objects to paths, and frames. A frame is a placeholder, much like a dummy or null object in other programs.

Behavior Modification Next, you want to make all of Davey's animations available to NeMo. Find them in the Resource view and drag them, one at a time, onto Davey in the 3D Layout. Then, to activate the various animations, add a behavior; here's where the fun really begins. NeMo offers a range of controller behaviors using one of the standard input devices: keyboard, mouse, or joystick. But one of its most versatile tools for character animation is the Unlimited controller, which lets you assign different animations to joystick and mouse actions in a list dialog (Figure 2). Choose the input from one drop-down list and any animation you've applied to the character from a second list. For each activity, you can assign a priority, whether and how an animation blends with others, a time base, and more.

NeMo's behaviors are in a resource window called Building Blocks, which contains a wealth of tools for creating interactive applications. Here's some of the Building Block categories with examples:

- 3D Transformations: Move To moves a 3D entity to a specified location over a set time or number of frames.
- Characters: Animation Synchronizer generates messages at specific times during animation playback.
- Collisions: Collision Detection provides the all-important function of sending a message when two objects touch.
- Controllers: Get Mouse Relative Position outputs the number of pixels the mouse has moved after stopping.
- Mesh Modifications: Skin Join lets you simulate seamless characters from jointed hierarchies.
- Particles: Useful for special effects such as fire, smoke, explosions, and rain. Particle systems come in various shapes, including cylinder, disc, sphere, point, and planar. You can also emit particles from a curve and an object's surface. You can apply texture images and animations to particles, as well as "interactors" for real-world effects such as air resistance, gravity, wind, and deflection. There's also a "mutation box" interactor that causes particles that enter its volume to mutate in size, color, and texture.

Once you've applied behaviors to an entity, you have to do some wiring, using the Link function in the Schematic view. (Take another look at Figure 1. The Schematic view is shown in the bottom half.) NeMo uses two types of links. Typically you start out with event links, hooking up the main output of a building block (on the right side) to the input of another (on the left side). For example, if you want a keypress to activate a character's animation, you can't use the unlimited controller, so you would use a keyboard controller that would wait for the desired keypress, and then pass an "activate" message to the animation block it's linked to. You can even link an event to itself, creating a loop.

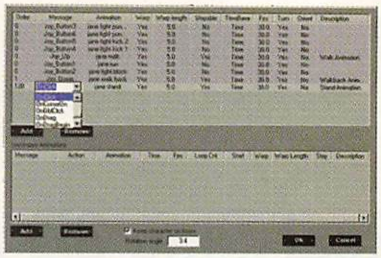


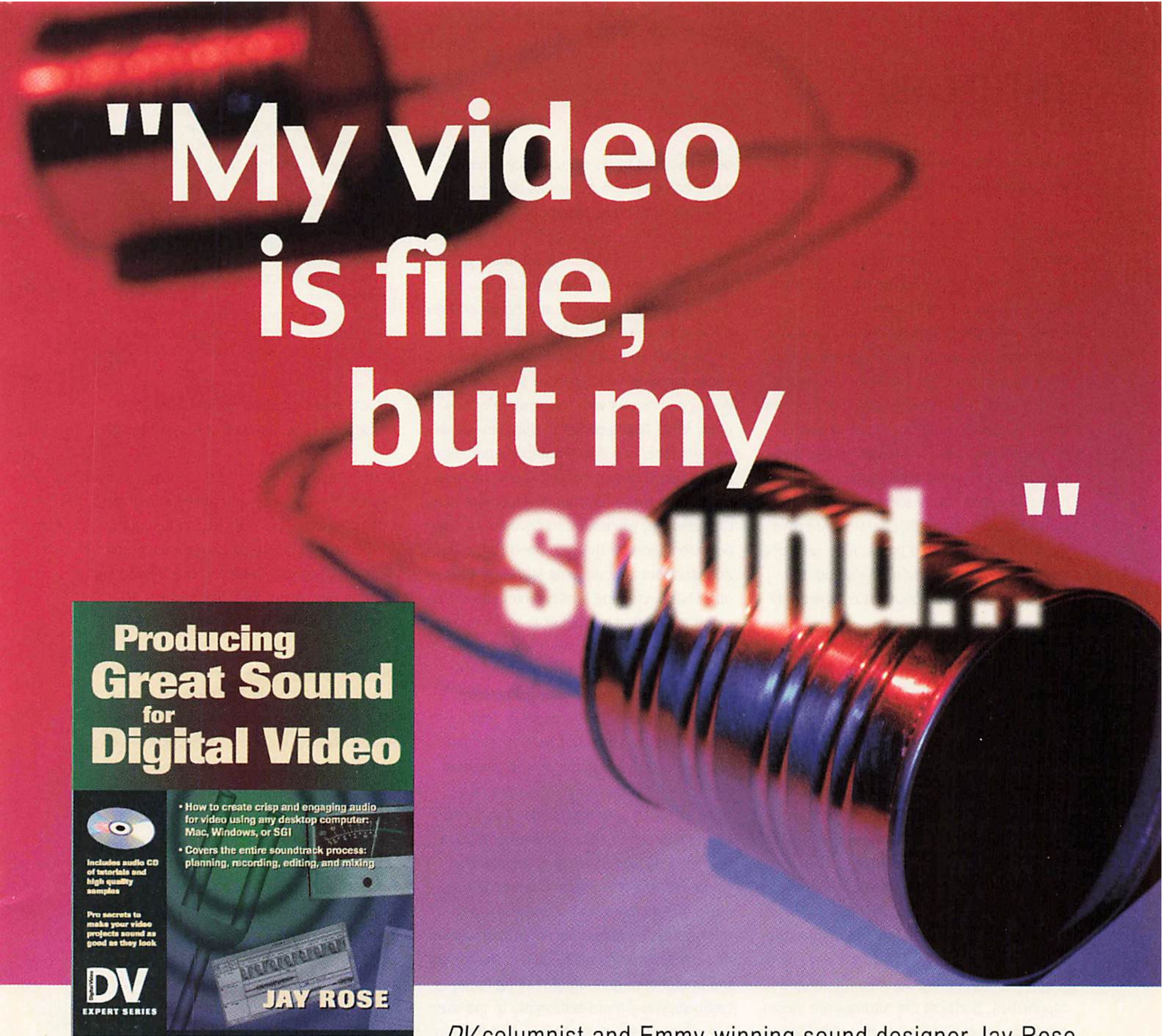
FIGURE 2. NeMo Creation's Unlimited controller lets you assign different actions to a variety of inputs.

The other type of link is a parameter link, which lets, for example, a function reduce a character's health setting when he's hit by a bullet. These are indicated by dashed lines, compared to the event link's solid line. NeMo's Schematic functions let you create a working flowchart of your program's logic, and add comments as well. You can "bundle" a section of your schematic into a "black box," exposing only those parameters that need to be accessed by other functions, and save it in a library to reuse elsewhere. Another nice feature is that you can edit the schematic even while your scene is live and see the results immediately. Programming was never like this!

In all, NeMo includes over 200 behaviors, all of which are useful and well designed, and more are being created by users and distributed freely. In addition, the program comes with a wide variety of demo files that you can load, examine, run, and modify. One particle demo uses a rotating fan to blow flames in different directions. Other demos show how to do collision detection, camera control, character animation, morphing, and dynamics.

Also related to program organization is the Level view, which is typically tabbed with the Schematic view in the bottom half of the screen (you can drag views to the three different UI areas, but you can't float or resize them). The Level view displays the composition contents as a hierarchical list and lets you move and copy scene elements, scripts, and so on by dragging and dropping. It's particularly useful for finding specific items in complex scenes.

When you've finished development, you use the Export to NeMo Viewer function, which produces a single file containing all program content and logic. This can be played with the free standalone NeMo Player or with a version of the player that plugs into Netscape or Internet Explorer. It can also be brought back into NeMo, but the program



Paperback with audio CD, ISBN 0-87930-597-5, $39.95

"A must have audio resource for anyone involved in video, multimedia or Internet production."
–Randy Cates, columnist for *Videography*

"Undoubtedly, this is a book that's destined to be THE essential work on sound production for project-video users." –Mel Lambert, Media and Marketing, Hollywood

"A crisp, tight, practical how-to book where production insights and experience just leap off the page at you. All focus, no fluff."
–Dave Moulton, author "Golden Ears" ear training course

DV columnist and Emmy-winning sound designer Jay Rose has heard all the problems of creating compelling audio for digital video. If your soundtracks aren't as crisp as what you hear on network TV, try Jay's pro techniques in ***Producing Great Sound for Digital Video***.

Jay reveals solutions to problems throughout the entire audio process—from pre-production to mix—and *why* they work. You get step-by-step tutorials, shortcuts and advice for making great tracks with any computer or software, plus an audio CD of examples and exercises.

Available at bookstores everywhere.
Or call (800) 848-5594.

MF Miller Freeman Books

6600 Silacci Way, Gilroy, CA 95020 USA
fax: (408) 848-5784
e-mail: mfi@rushorder.com
web: www.books.mfi.com
Distributed by Publishers Group West

DV

DV2000

logic isn't available in Schematic view; thus your algorithms are protected.

Interaction & Documentation European software developers have their own unique ways of doing things. Some might call them individualistic, others eccentric. NeMo is a classic example of this syndrome. It's a Windows program, but it doesn't adhere to Windows conventions. For example, the program screen when maximized covers the Windows task bar, even if the latter is set to Always On Top. And its task bar button doesn't have an icon. When you're tabbing through a multi-field dialog, the cursor sometimes disappears for a bit, then reappears if you press the Tab key a few more times. Also, lack of a quad display can hinder scene setup. But the program is quite robust; it crashed only once in my extensive testing.

NeMo's weakest point (and this is a biggie) is its documentation. It's a deep, rich piece of software, but the reference material is sketchy. The reference documentation window works like any other program window, so it's hidden behind any open dialog. Also, it's in HTML format and uses a nonstandard structure, with a limited table of contents and no index. Worst of all, there's no search capability, so looking something up can be a bit like searching for the proverbial haystack-bound needle—frustrating as hell.

On the plus side, the printed manual provides a nice range of tutorials but not nearly enough to teach the program's full range of capabilities. Some of the tutorials are incomplete, so you can't follow them unless you do some digging. Despite the documentation's weakness, NeMo is eminently usable, as demonstrated by FunRun, a decent racing game completed in 10 weeks by German development house Terratools (Figure 3).

It's All Fun & Games... If you think you're going to sit down with NeMo and create a game single-handedly in a few hours or even a few days, think again. You still need most of the skills and concentration of an experienced producer of interactive software to produce good games with this system or any other. What NeMo brings to the game-development table is a visual authoring paradigm, plus a bunch of general behaviors you may be able to use to shorten development time. Chances are, you'll want to develop your own behaviors, and for that you need NeMo Dev and classical programming skills. On the other hand, creating a highly interactive 3D presentation that's accessible to anyone on the web is much easier with NeMo, where it would be quite difficult with anything else.

Is the typical NeMo user going to create the next *Quake* or *Battlezone II* or *Tomb Raider*? Probably not, but he or she could produce an original game or web diversion and have a fair bit of fun in the process while still getting to have a life. And with a bit of inspiration and a whole lot of elbow grease, a driven individual or small group could use NeMo to create something that could make money and cause gamers to think a little bit differently about themselves and the world. With all its quirks, NeMo has the overwhelming virtue of making complex technology way more accessible to the average person than it was before, and that's a good thing. And as soon as the program comes with adequate documentation, including lots more tutorials and a searchable online reference, it'll be a very good thing.

nemo creation 1.0.1

NeMo Creation 1.0.1, due by the time you read this, will have improved documentation, plus:

- Support for Windows NT (1.0 works only in Windows 9x)
- 3D sound engine based on Aureal A3D
- New editors (mainly for setups, such as sound and 2D sprite setup)
- New behaviors
- Players for Director and web browsers.

Further down the road, NeMo plans to offer cross-platform player support and improved web features such as streaming playback.

Virtus OpenSpace 3D Author If you've ever used a VRML authoring tool like Paragraph/Cosmo's HomeSpace Builder or Platinum's VRCreator (both companies/products long defunct), you won't have any trouble coming to grips with Virtus' new OpenSpace 3D Author 1.0 (OSA). It's easy to populate a 3D scene by dragging geometry from the Catalog window at the bottom of the interface into the 3D view above it. The program comes with an assortment of primitives and animated objects (a door, a desk), plus a nice selection of objects from the Viewpoint library. Also, you can import objects in AutoCAD (.dxf/.dwg) and .3ds formats. The catalog also provides materials (image files) that you can drag onto objects, as well as cameras, lights (directional, point, and spot), .avi movies, and .wav sound files. Alas, unlike its predecessors, OSA doesn't show thumbnails in its catalog, so you've got to load something before you can see what it looks like.

Once you have objects in the workspace, you can transform them using a handy all-in-one "manipulation frame," shown enclosing the bicycle model in Figure 4. Drag the red

FIGURE 3. Creating a racetrack for a NeMo game in 3DS MAX.

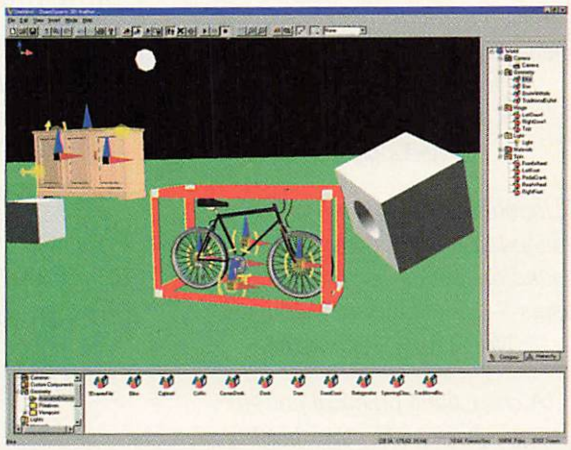


FIGURE 4. OpenSpace 3D Author provides an intuitive gizmo for transforming objects.

bars to rotate, the white corners to scale, or anywhere else to translate. You can do transforms in any of three coordinate systems: world space, the current camera's space, or the current object's local space. You can also navigate through the scene in one of three different modes: Walk, which translates the point of view in relation to the screen center; Fly, which translates the point of view in relation to the camera's current orientation; and Orbit, which rotates around the scene while remaining pointed at the center. Meanwhile, the status bar at the bottom of the screen provides helpful statistics such as frame rate, total number of polygons, and the number drawn in the current frame. There doesn't seem to be any way to determine the polygon count for a specific object, though.

Most commonly used functions are handily provided in a toolbar of icon buttons, right below the menu bar. Other UI elements include two tabbed Scene Information panels: one that lists the scene contents hierarchically by category and one that lists them alphabetically. The latter mode lets you access members of object hierarchies so you can animate them independently of the entire object. If Orbit mode is active, selecting an object from the list centers it in the view because it becomes the center of the scene. This can be handy but also a bit disconcerting if you're zoomed in. Fortunately, a Home function lets you quickly re-orient yourself.

Despite its similarity to VRML authoring tools, OSA doesn't export to that format. To distribute your presentation, you save it in the OpenSpace native format (.o3d), making sure you've set the program to embed textures in the file (audio files can be external). Transfer it to the recipient, who must have the OpenSpace 3D Player program—a free 5.7MB download from Virtus' web site (www.virtus.com) that works as a standalone program and as a plug-in to the Netscape and IE browsers.

What about building interactivity? At the base level, the player program lets the viewer navigate through the scene using mouse and keyboard commands, as in a VRML browser. If you need a bit more, you can create animations of varying complexity. The easiest method is to drag one of OSA's built-in animations onto an object. These include various simple hinge, spin, roll, slide, and move animations. You can set them to run or not upon starting playback and optionally let the viewer right-click an object to toggle its animation. Other options, set from a dialog, include movement speed and direction and three different sounds that play on activation, deactivation, or while animated. You can also drag an HTML link onto an object and specify a URL.

sources

NeMo Creation • list price $990, one seat
NeMo SA (formerly Virtools)
Digimation (U.S. distributor)
(800) 854-4496
www.nemo.com • www.digimation.com
System Requirements:
Pentium (PII recommended) 200MHz or faster, Windows 95/98/NT 4; 48MB RAM; 60MB hard disk space; Direct3D or OpenGL compatible 3D accelerator card.

OpenSpace 3D Author • list price $395
Virtus Corp.
(919) 467-9700 • www.virtus.com
System Requirements:
Pentium 200MHz or faster, Windows 95/98; 64MB system RAM; 50MB hard disk space

To get more customized animation, you can set an object to move along a path. Path setup is complex and awkward; I won't bore you with the details. Suffice it to say, this aspect of OSA was apparently designed by someone lacking even the slightest clue as to the meaning of user-friendliness. Part of the process uses the Scheduler, which essentially provides a scripting interface to OSA.

If you're not averse to programming, you can use the Scheduler to build reasonably elaborate interaction. For example, when the user enters a room, play a certain piece of music for five seconds, then stop. The scripting language is an extension of Lingo for Director. OSA was originally designed as a 3D-authoring add-on for Director. It includes triggers, which cause actions to happen. An action contains one or more rules, which are performed in order. Each rule can do one of three things: set a property such as light intensity, send a message such as telling an object to begin rolling, or execute a command such as Wait. Essentially, you're programming the commands available via the GUI. The language provides no additional functionality.

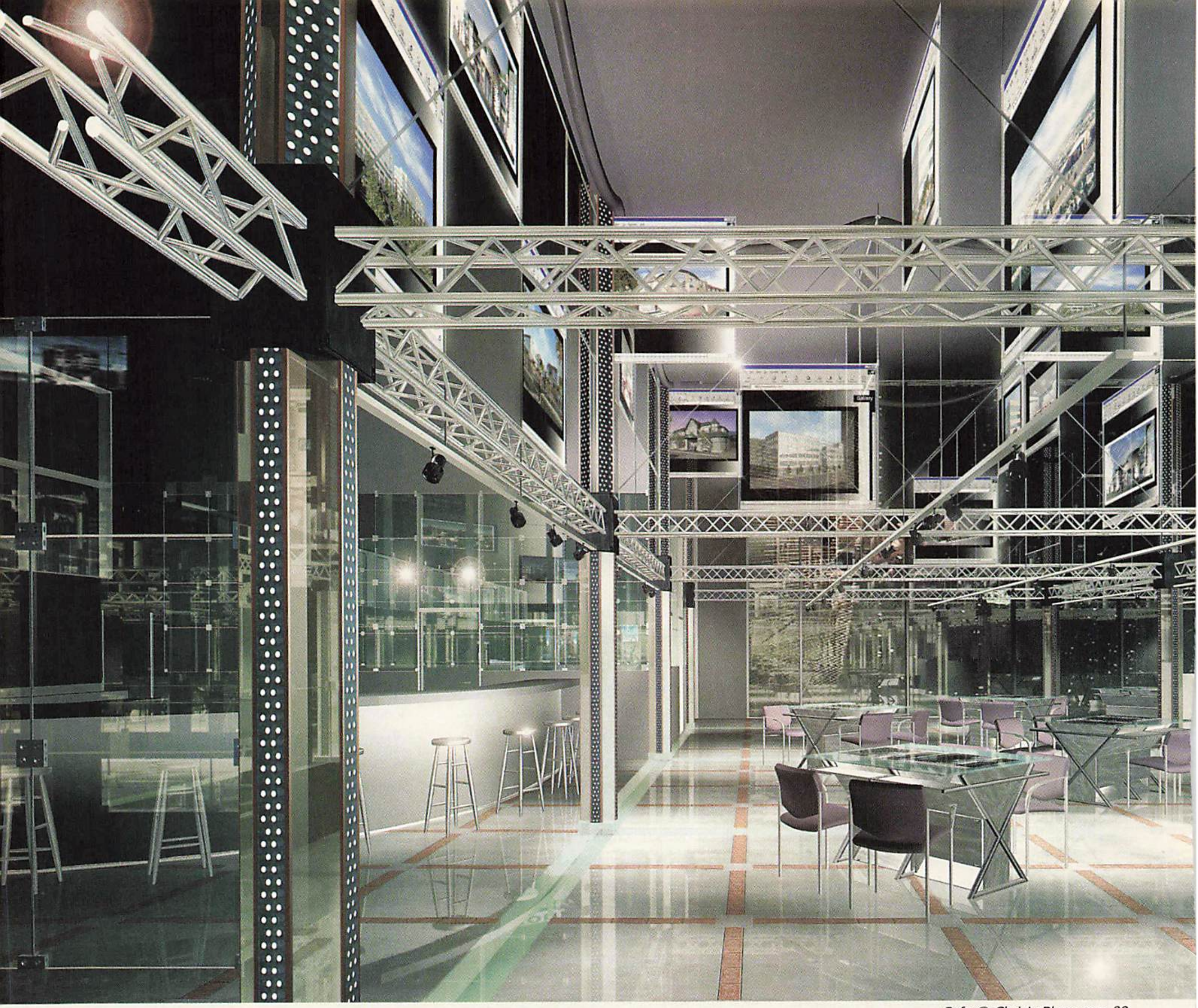
Gripes & Drawbacks Overall, OSA works reasonably well, but it has some annoying bugs and shortcomings. A couple of times, it crashed immediately after loading. One imported object would disappear and reappear each time I clicked on it, and another got "stuck" in Navigate mode, so that I could no longer transform it. Another time, I tried to set the program to Top view, and everything disappeared. After switching views a few more times to try to figure out if I was doing something wrong, it crashed. Every time you add an object to the scene from the library or apply a material to an object, the screen flashes annoyingly. These and other display problems are apparently related to OSA's OpenGL implementation.

Because there's no quad display letting you view the scene from different angles simultaneously, you have to change the view often, switching among Top, Right, Front, and Camera views. The program offers keyboard shortcuts for the first three, but none for the Camera view. In Windows NT, the authoring program supports sound during playback of scenes, but the player doesn't.

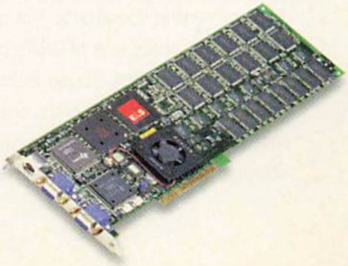
OSA suffers from the all-too-common disease of one-directory-for-all-files-itis. If you import an object from a CD, then use Save As to store the scene, it goes to the same directory. Similarly, if you import a file that uses a bitmap file as a texture, it asks you for the location of the bitmap and applies it to the object using the original mapping. What's bad is that, if you want to import another object from the same directory as the previous one, you must navigate back to the original directory, then back to the map directory, and so on. Speaking of maps, the program supports a nice variety of file types, including .png, .jpg, and .tga, but oddly enough, it doesn't support .gif. Even more oddly, if an imported texture uses a .gif image, the program doesn't blink but still tries to load it.

The Long & Short of It Despite their shared function as 3D authoring tools, OpenSpace 3D Author and NeMo Creation don't compete head-on. OSA is much easier to use and lets you set up and sequence animations fairly easily, but it doesn't provide interaction in the true sense of the word. It's okay for a product presentation or architectural walk-through, but you couldn't use it to design a game. Unfortunately, because of its lack of stability, I can't recommend it. NeMo, on the other hand, suffers from a slightly bizarre user interface but has plenty of interactivity-building power under the surface. It's usable with some extra effort and, with improved documentation, could be a winner.

David Duberman is a technical writer specializing in 3D graphics and animation, based in the San Francisco Bay Area. Contact him at duberman@dnai.com.



Cafe © ChrisLeBlanc.com 99



BY RAF ANZOVIN

Secondary Actions Make the Character

Tails, ears, clothes, and hair move indirectly, but have a life of their own.

The Cat is not able to control all her limbs in this shot from *The Animator's Apprentice*.

Movement that results indirectly from a character's main actions is called secondary action. This includes the movement of hair, clothing, floppy ears (think Jar-Jar Binks), and other passive objects that hang off or drape on the character. Because floppy ears, long hair, capes, and the like are not rigid or rigidly fixed, they will only move in reaction to the main line of movement (because of Newton's First Law, the one about inertia). That is, they will tend to lag behind the main action, keep going when the character comes to a sudden stop, and otherwise act like objects governed by real-world or cartoon physics.

Secondary action also includes parts of a character—arms, fingers, tails, fins, and so on—that are influenced by the main action while being under the character's conscious control. For instance, the hand may trail behind the arm's line of action but still have a motivated, nonpassive motion. It may, for example, culminate in a deliberate snapping of the forefinger. Clothing isn't necessarily passive either. Its secondary motion can be used to good effect as an extension of the character's personality, especially in more cartoony styles of animation.

Many programs now offer procedural-dynamics tools that can be used to automate secondary actions, at least to some degree. These tools are great for creating passive secondary actions in cases where the animator would essentially be acting as a human physics simulator—for example, when animating fur or clothing. But a procedural system is rarely of much use for active secondary actions, which need to express motivation and not merely reaction. In this column, we'll look at approaches to animating secondary actions entirely by hand.

As an example of mixing active and passive secondary action, we'll examine a shot from *The Animator's Apprentice* that appears a bit later than last month's shot ("Pose-to-Pose, One Sequence at a Time," January 2000). In this sequence, the Cat totters down two stairs, first looking at her own feet with puzzlement, then looking at the camera (Figure 1). The most important thing conveyed in this shot is that the Cat's main actions are not under her own control. Dennis the Dog, offscreen, has taken control of the cat's legs with his magic animator's gloves and is forcing her to step forward by remote control.

The Trail of the Tail First, we'll look at the animation of the Cat's tail. In this shot, her tail hardly has any volition, so the motion it requires is a simple whip motion. Whips are used in most cases of passive secondary action where one end of the object is anchored. Figure 2 shows what's going on. As the Cat's body is moved to the side, the base of the tail is dragged with it, but the tip stays where it is. Then, while the Cat holds for a few frames, the tip of the tail has a chance to catch up with the base and whips past it. When the Cat moves to the other side, the tip once again lags behind the base, quickly overshoots it, then comes back to rest in a curled position that's not quite as extreme as the overshot position.

The exact timing and extent of the whip motion has a lot to do with the length, rigidity, and tensility of the object—a tail whips farther and with more rebound than a whisker—and you must also consider whether the object is of the same rigidity

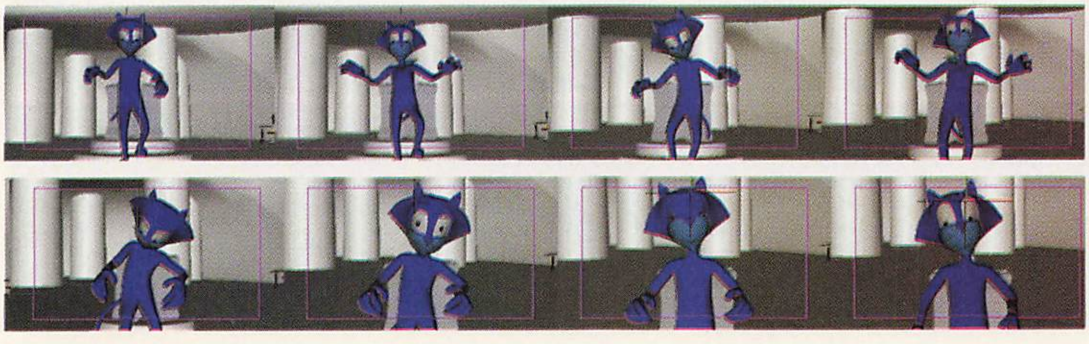


FIGURE 1. As the Cat steps toward the camera, a lot of active secondary action is used on her upper body to make it clear that her motions are not entirely under her control.

FIGURE 2. The Cat's tail, an example of passive secondary action, is supplied with a whip-like motion as the tip of the tail lags behind and then overshoots the base.

along its whole length. A cat's tail is extremely flexible along almost all of its length, and thus can curl and whip in any direction. Dog tails are comparatively rigid, flex primarily at the base, and curl up along the vertical axis only.

Note that while there's a lot of forward motion to the Cat as she takes steps, it doesn't have much of an impact on the tail. Looking at the tail of a real cat or dog, it's clear that forward motion alone doesn't produce much effect on the tail, while motions up and down or to the side cause pronounced changes.

Heads & Arms Let's look at the animation of the Cat's spine and head. Normally, the head of a character leads any motion the character makes, but in this case, the head and torso have been given a lot of secondary action to sell the idea that she is uncoordinated and not motivating her own movements. Figure 3 shows a side view of the head and torso in motion. In some ways, their movements are similar to the whip action of the tail but toned down considerably, since the cat is trying to exercise more control over her head in order to look at her feet. As the Cat is jerked forward with each step, the head and torso trail behind for a moment, then snap back to a forward position as the Cat comes to rest. Their movements are offset, so that the head reaches its backwards position several frames after the torso does. This produces an interesting rolling movement that makes the head and shoulders appear loose and uncoordinated.

The arms themselves do not move much. Most of the secondary action is in the hands and fingers. The hands drag behind every up-and-down movement; the fingers drag behind the movement of the hands, with an offset so that they appear to "roll" in the same manner as the Cat's head.

Fully Vested Movement For an example of purely passive secondary animation involving clothing, let's look at a shot in which I animated Dennis's vest by hand (Figure 4). Most of the shots currently in production are animated with Hash Animation:Master 2000's new cloth-dynamics tools, but these weren't available when this shot was animated.

Animating clothing by hand is particularly challenging. Not only must the cloth appear to move in a believable way—taking into account gravity, the character's momentum, any wind or other external forces, and the cloth's own characteristics (thick or thin, stiff or flimsy)—it must also appear to fit closely around the character without ever going through its surface. Figure 5 shows the vest's bone structure, which gives enough control to wrap the vest around Dennis's body during difficult movements. Sometimes the vest's position had to be cheated; it was often allowed to go through his back and sides, as long the penetration was hidden from the camera.

The general rules for movement of the vest were:

- In the frame where Dennis hit a particular pose, the vest is stretched out behind him toward his previous pose.
- About five to 10 frames later, it settles against his body (taking gravity into account so its flaps hang downward most

FIGURE 3. The Cat's torso, head, and arms respond with secondary movement as she moves forward. Note that the head movement is offset from the torso, so that it reaches its extremes of movement a few frames after the rest of the upper body. This produces an interesting rolling motion.

FIGURE 4. The vest of Dennis the Dog was hand-animated in this shot. As he reaches each pose, the vest trails behind him and settles against him in five to 10 frames.

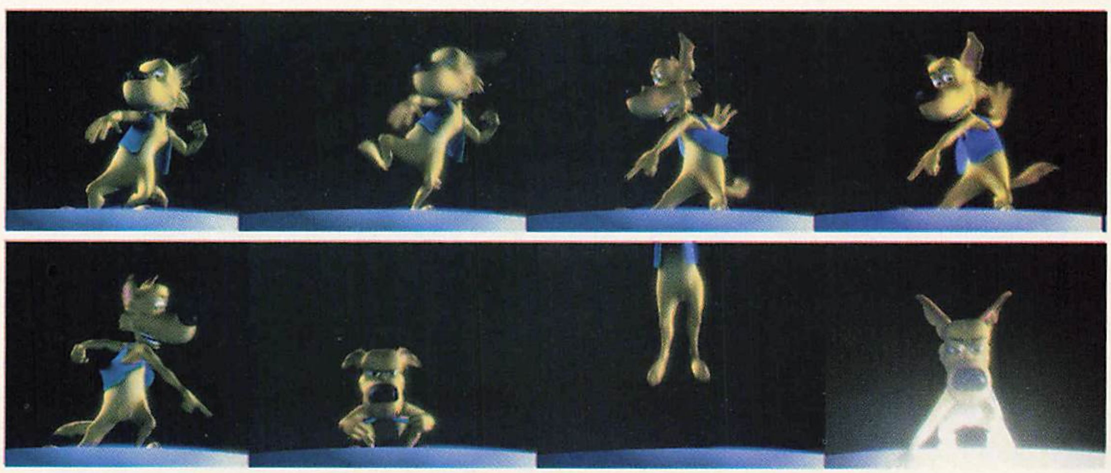


FIGURE 5. The vest's bone structure provided enough control to wrap it around Dennis's body successfully even in his more extreme movements.

of the time). More frame delay is used when Dennis's movement is quick or extreme.

- The vest holds position until his next move, when it trails behind him again.

In this shot, the vest begins with the right flap settled against his body and the left flap hanging down. When Dennis hits his next pose, both flaps trail behind him. About five frames later, they both flap forward and curl around him.

Every shot required some elaboration of the basic precepts because each flap of the vest moved somewhat independently, and in several instances it was necessary to vibrate the vest back and forth or do other unusual things to make it fit with Dennis's fast movement. At the end of this shot, for instance, he leaps into the air. This required that the vest be slicked back against him for the duration of the upward movement of the jump, and then suddenly snap upward and remain there as he came back down.

As procedural dynamics systems mature, animators won't be doing as much of this type of work in the future (I know I don't plan to). But you need to understand the physical characteristics and motivations of the character you are animating before you can set up a dynamics system properly, and one way to gain that understanding is by hand-animating secondary motion. We'll take a look at what dynamics can do for you in a future column. Next month, we'll start a series on setting up and animating faces.

Raf Anzovin is the cofounder of Anzovin Studio, a character animation house based in Amherst, MA. Contact him at raf@anzovin.com.

PIXELMONKEY *continued from p. 57*

I have a collection of brushes for specific uses. For example, a few have little scratch marks in them, designed to add scratches to glass. I've also seen people make brushes of hand prints and fingerprints so they can add that extra touch of realism in their specular and reflection maps. Having a standard arsenal of specialized brushes will save you piles of time.

In addition to painting, experiment by using brushes as eraser tools. Figure 4 is an example of a map I created in about three minutes using my brushes as painting and erasing tools.

Good brushes can save you hours of time and make your job much much easier. After a few months of experimentation and hard work, you'll have a set of killer tools in the tool chest. Until next time, paint on!

Robert Nederhorst is a pixel coordinator and rendermonkey #3 at Digital Domain. Contact him at throb@d2.com.

BY ROBERT NEDERHORST

Custom Brushes for Photoshop

Do-it-yourself brushes to make your textures better.

Back again, eh? This month we're going to cover something lots of beginners—and some professionals—overlook: custom brush sets. This excellent Adobe Photoshop feature lets you make a brush based on a piece of an image or even from another brush. This greatly expands your range of texturing tools, which makes your maps more interesting, which makes your scenes better. It's a quick study, so let's jump right in and make some cool, useful brushes.

To access the brushes in Photoshop, select the airbrush (or paintbrush or eraser) and go to Window→Show Brushes. Your brush palette will appear. If you're like most people, you'll have the standard brushes with different sizes and softness values. These are good, but we need more painting power. You can create a brush from a pre-existing image. Here's a quick and easy one to demonstrate the technique, then we'll apply it.

Let's create some text (I'm using "TEST") as a starter. Try black text on a separate layer from the white background. Select the entire background layer and delete all the stuff inside it but not the layer itself. We'll make a separate layer to keep the parts we want away from all the nasties we don't want in our brush. Now, using your favorite selection tool, select the text (on its layer). Keeping the text selected, go to the brushes palette and click the arrow on the upper right-hand side. Select the Define Brush option. If you look at your palette, there is a new brush in it. Now use that airbrush and pick your newly created brush. Paint away!

OK, so I have gotten you to make an absolutely boring brush. My apologies. Get rid of that awful thing by right-clicking on it in the palette and selecting Delete Brush. Let's load an image we can use to create a cool brush to paint with. I'm going to a site we've used before, Jeremy Engleman's at www.art.net/~jeremy and using the carport image, carport.jpg (Figure 1). Load the image into Photoshop and change it to grayscale. Adjust the brightness and contrast to get the values you want for a cool-looking brush (Figure 2). Using your favorite selection tool (in my case, it's the lasso tool), select the area you want to make a brush from (Figure 3). Depending on how you want the brush to look when you paint with it, you may or may not choose to feather your selection. A non-feathered selection will create a harder-edged brush than a feathered selection. After you have made your selection, go to the brushes palette and define your brush as before. Voilà! You have (hopefully) made a good custom brush. Now create a new image and paint using your new brush. Like it? If not, create more!

To create more, you can either grab pixels off an existing image, or you can get even trickier. Lots of times, I'll make a brush and paint with it a bit to define another area that (you guessed it!) can be used for another brush shape.

To get you started, download my brush file at www.3Dgate.com. Load the .abr file into your brush palette by choosing the Load Brushes option in the Brush Palette menu (the one with the arrow). This palette includes the brushes I use on a regular basis. If you have a good set of brushes you use all the time, save it and take it with you if you work on another machine. You can do a Replace Brushes anytime, on any file, so you always have your favorite brushes wherever you go.

Continued on p. 55

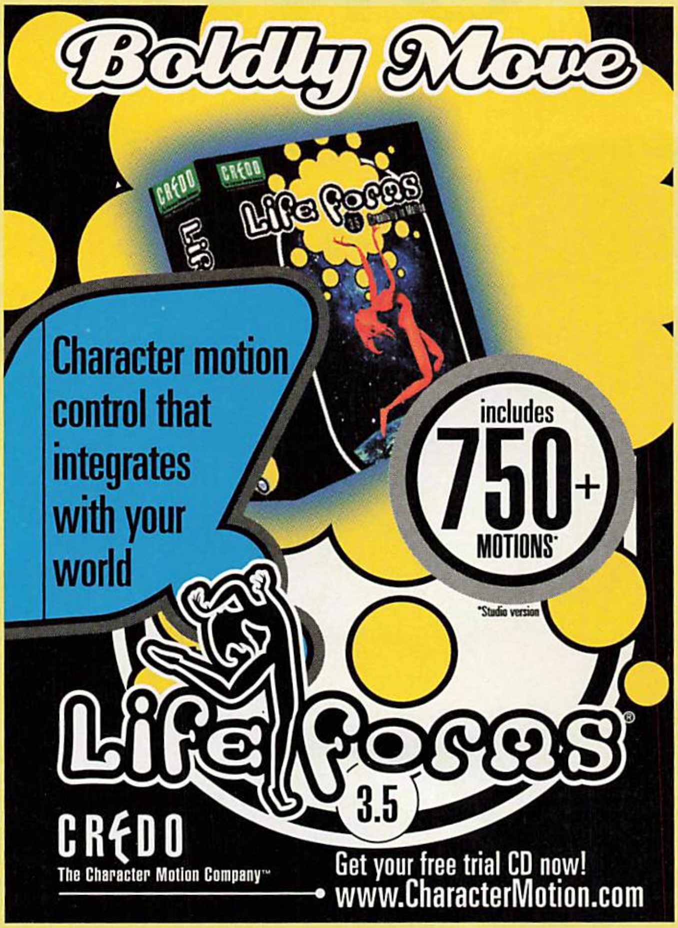
FIGURE 1. Our brush source image.

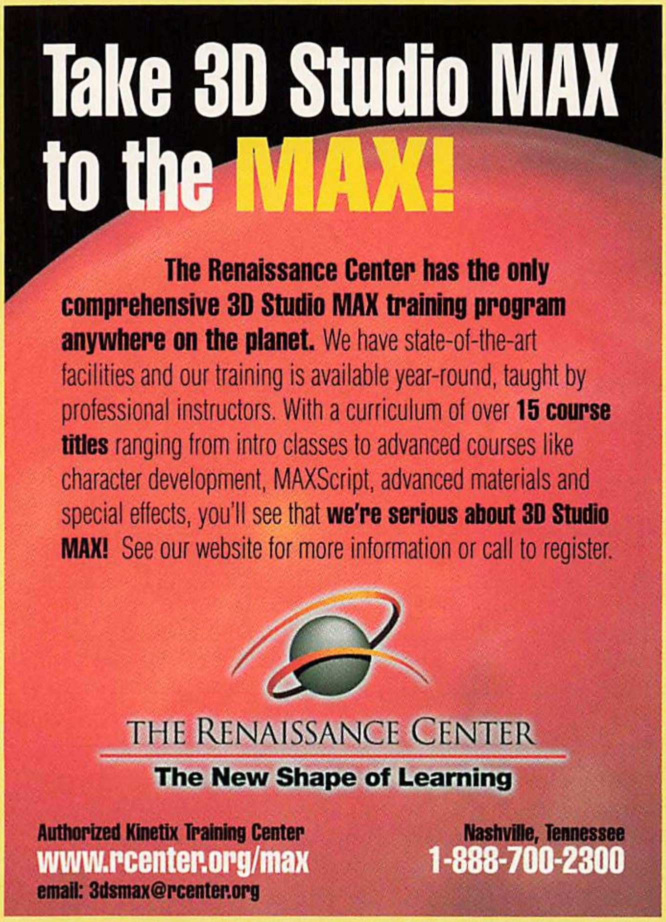
FIGURE 2. The same in black and white.

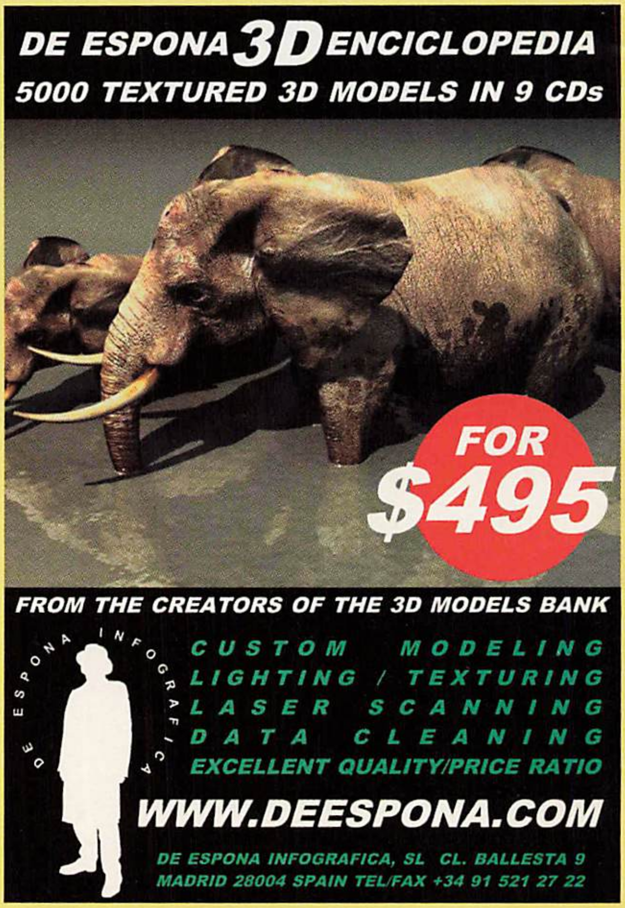
FIGURE 3. The area I selected to be the brush.

FIGURE 4. Using brushes as painting and erasing tools, I created this map.


The Most Complete Solution for Professional 3D Scene, NURBS and Animation Translation!
PolyTrans v3™
With Maya® & MAX® Plugin Support!
Used worldwide by thousands of 3D-industry professionals, PolyTrans is an indispensable production pipeline tool that performs clean, robust and reliable translations plus interactive viewing of the most popular 3D file formats. See for yourself by downloading a demo today!
3D Studio® r4
3D Studio® MAX §‡
Apple® 3DMF®
Alias® triangle
Cinema-4D®
Detailer®
DirectX®
DXF®
EI®, Form/Z® FACT
GE2 (Mirai®)
Houdini® GEO ‡
IGES 5.3 ‡
Imagine®
Lightscape®
Lightwave®
Maya® §‡
Minolta Vivid® 700
OpenGL C Code
OpenFlight® (FLT)
POV 2 & 3
Pro/Engineer® .slp
Renderman® RIB ‡
Renderware® RWX
SoftImage® hrc/dsc ‡
SAT 5.3+ (ACIS®)
Solid Edge® V4-V7+ §
Solid Works® 99+ §
Stereolithography STL
Strata® StudioPro v1.75
trueSpace® 2+
USGS & Gtopo30 DEM
VistaPro® DEM
VRML™ 1 & 2
Wavefront® OBJ/MTL ‡
Some formats only import or export, or available on Intel only. ‡ = NURBS. Red = New in V3.
§ = Requires resident copy of this application.
www.okino.com for file format list, demos, testimonials & user list
Import Converter Modules
N-Sided Polygon Meshes With Holes
Vertex Normals, Vertex Colors, uv Textures, UV Tangents
Trimmed NURBS Surfaces
Object Hierarchy and Instancing
Animation Channels
Lights, Cameras & Related Attributes
Materials and Shader Attributes
Texture Mapping & Attributes
Texture Projections
Core PolyTrans Functionality
Animation Storage, Playback & Interpolation System
Hierarchical 3D Database Management System
BDF File Transfer Format Save and Load
Animation Keyframe Resampling & Reduction System
Trimmed NURBS to Polygon Conversion and Optimization
Okino's Photo-Realistic Scanline Renderer
Real-Time Object Collision Detection System
Polygon to Polygon Mesh Optimizer & Processing System
2D Bitmap File Conversion Sub-System
Export Converter Modules
Windows NT/98/95 & SGI IRIX
Original, Lightwave
3D Studio MAX
A|W's Maya/NT
SoftImage|3D
MultiGen's Creator
Nichimen's Mirai
Cinema-4D
trueSpace
Renderman
Mechanical Desktop
Toll Free: 1-888-3D-OKINO, WEB: http://www.okino.com
T: (905) 672-9328, F: 672-2706, Email: sales@okino.com
All products mentioned are registered trade names of their respective holders. "apeBot" scene copyright © 1999 by Newtek, Inc. and Matt McDonald of Vision Scape Imaging, Inc.
Okino Computer Graphics


Boldly Move
Character motion control that integrates with your world
includes 750+ MOTIONS*
*Studio version
Life Forms® 3.5
CREDO
The Character Motion Company™
Get your free trial CD now!
www.CharacterMotion.com


Take 3D Studio MAX to the MAX!
The Renaissance Center has the only comprehensive 3D Studio MAX training program anywhere on the planet. We have state-of-the-art facilities and our training is available year-round, taught by professional instructors. With a curriculum of over 15 course titles ranging from intro classes to advanced courses like character development, MAXScript, advanced materials and special effects, you'll see that we're serious about 3D Studio MAX! See our website for more information or call to register.
THE RENAISSANCE CENTER
The New Shape of Learning
Authorized Kinetix Training Center
www.rcenter.org/max
email: 3dsmax@rcenter.org
Nashville, Tennessee
1-888-700-2300


DE ESPONA 3D ENCICLOPEDIA
5000 TEXTURED 3D MODELS IN 9 CDs
FOR $495
FROM THE CREATORS OF THE 3D MODELS BANK
DE ESPONA INFOGRAFICA
CUSTOM MODELING
LIGHTING / TEXTURING
LASER SCANNING
DATA CLEANING
EXCELLENT QUALITY/PRICE RATIO
WWW.DEESPONA.COM
DE ESPONA INFOGRAFICA, SL CL. BALLESTA 9
MADRID 28004 SPAIN TEL/FAX +34 91 521 27 22

EDITORIALresources

3D

Voodoo4
Voodoo5
3dfx Interactive Inc.
(888) 367-3339
www.3dfx.com

3D-Doctor 3.0
Able Software
(781) 862-2804
www.ablesw.com

Photoshop
Adobe Systems Inc.
(408) 536-6000
www.adobe.com

Maya 2.5
Maya Fur
Composer
Alias|Wavefront
(416) 362-9181
www.aw.sgi.com

Amapi 3D Tutor Interactive CD-ROM
Animhouse
www.animhouse.com

form•Z 3.0
auto•des•sys
(614) 488-8838

SceneGenie
SceneGenie Limited
Autonomous Effects Inc.
(650) 934-0131
www.afx.com

Softimage 3D
Avid
(800) 949-2843
www.softimage.com

GenShade
Cinema Graphics
(818) 718-6320
www.cinegrfx.net

Shag: Hair
Shag: Fur
Digimation
(800) 854-4496
www.digimation.com

3D Forest CD-ROM
Digital Forays
www.3dforest.com

dpsReality
Digital Processing Systems
(606) 371-5533
www.dps.com

3D Studio MAX 3.0
Discreet
(415) 547-2000
www.discreet.com

Rust and Metal
Concrete and Stone
Skin Textures CD-ROMs
Dosch Design
+49 9391915853
www.doschdesign.com

Erazor X
ELSA
(408) 961-4600
www.elsa.com

Trooper
Endless Corp.
(323) 962-5656
www.endless-corp.com

Mammoth-2
Exabyte Corp.
(303) 442-4333
www.m2wins.com

FAMOUSfaces 1.5
FAMOUS Technologies
(415) 835-9445
www.famoustech.com

FurShader
Glassworks Ltd.
www.glassworks.co.uk

Spaceball 4000 FLX
Labtec Inc.
(360) 896-2000
www.labtec.com

Cinema 4D
Maxon Computer Inc.
(805) 494-0718
www.maxon.net

FiberFactory2
MetroGrafx
(719) 748-3972
www.metrografx.com

Microsoft
(425) 882-8080
www.microsoft.com

NeMo Creation 1.0
NeMo SA
Digimation (U.S. distributor)
(800) 854-4496
www.nemo.com,
www.digimation.com

LightWave 3D
NewTek Inc.
(210) 370-8000
www.newtek.com

Mirai 1.0
Nichimen Graphics
(310) 577-0500
www.nichimen.com

GeForce
nVidia
(408) 615-2500
www.nvidia.com

Fur KIT
ParticleSuite
Phoenix Tools
+39 0267075747
www.phoenixtools.com

RenderMan
Pixar Animation Studios
(510) 236-4000
www.pixar.com

Stitcher
REALVIZ SA
+33 0492388460
www.realviz.com

Diamond Monster Shades
S3
(408) 588-8000
www.s3.com

Silicon Graphics Inc.
(650) 960-1980
www.sgi.com

MediaStorm II
SupremeGS
(303) 347-1940
www.supremegs.com

Amapi 3D 5.0
TGS Software
(858) 457-5359
www.tgs.com

OpenSpace 3D Author
Virtus Corp.
(919) 467-9700
www.virtus.com

Hair
Peter Watje
www.max3dstuff.com

ADVERTISERindex

ADVERTISER	PAGE
3D Conference & Expo	4-5
3Dlabs	8
Academy of Art College	63
Alias|Wavefront	12
ArchVision	63
ArtBeats	61
Autonomous Effects	61
Caligari	19
Creation Engine	63
Credo Interactive	60
De Espona Infografica	60
Dell Computer	C2-1
Evans & Sutherland	52
IBM Corp.	15
Immersion Corp.	21
Intergraph Corp.	C3
Journey Education Marketing	61
LIPSinc	55
MetaCreations	38-39
NAB 2000	56
NewTek	11
Okino Computer Graphics	60
Play Inc.	C4
Renaissance Center	60
Softimage/Avid	6
Technical Animations	61
Tiburon Entertainment	63
Vancouver Film School	61
Web Design & Development	58-59

The index on this page is provided as a service to readers.
The publisher does not assume any liability for errors or omissions.

BY CHRIS TOME

57 Channels & Nothing On

Almost a decade ago, I ordered a demo video from a little-known company named NewTek, which made a product called the Video Toaster. At the time, I was getting into professional computer graphics full tilt, and the Video Toaster, with all its capabilities, seemed like a killer deal. It even came with LightWave, which, after a PC port from the Amiga platform, would eventually become one of the most widely used 3D programs ever. While the Video Toaster went on to make NewTek a household name in CG circles, there was one particular thing on the tape that I'll never forget.

The video included an interview with NewTek's founders, Paul Montgomery (RIP) and Tim Jenison, in the late '80s. When asked about the idea of 500-channel cable systems (which are still today being promised as forthcoming), Jenison hypothesized that the number of quality programs would increase, but you'd have to weed through an exponentially larger amount of garbage to find them. That statement stuck with me and is a constant reminder that, just because you build it doesn't mean they'll come.

I recently watched a 3D-animated television show that shall remain nameless. Although I typically follow the "if you can't say something nice..." rule, this show was so ridiculously pitiful that it may have negative repercussions for a while on the broadcast 3D industry. Think of what happened to the Hollywood special-effects industry after *Tron* came out. That film may have singlehandedly caused five years of serious aversion to computer-generated special effects on the big screen. Now, of course, the industry is more mature, and one flop alone can't derail the 3D industry, but it can definitely still hurt.

Every summer it becomes increasingly clear that Hollywood hasn't got a creative clue when it comes to original content. Take this issue's *Stuart Little* cover story, for example. It's a rehash of an old book, albeit with a new 3D twist. (For the record, I haven't yet seen the film.) Ever play the game *Doom*? Well, Hollywood is making a movie based on the game, and how intriguing do you think it will be? As content creators, we entertain, inform, and even educate in some cases, and we should do it in original ways. Hollywood is hard-pressed for creativity right now.

As content creators, we entertain, inform, and even educate, and we should do it in original ways.

A vision long held by Jenison is that everyone should be able to create their own TV programs, and the industry as a whole is bringing us closer to realizing that dream. With the availability of inexpensive DV cameras, PCs that cost less than many home stereos, and 3D software proliferating beyond bleeding-edge artists, we're on the verge of a cultural renaissance. Nowadays, people with little more than the desire to create original content can not only do so, they can change the way Hollywood thinks about everything. Take *The Blair Witch Project,* for example. In my opinion, *Blair Witch* was a boring, utterly unenjoyable film experience (whoops—if you can't say something nice...), but it made Hollywood take notice of many things, not the least of which was how powerful the Net can be as a marketing vehicle. When something like *Blair Witch* comes along, people take notice because it's different. As a culture, we've become so jaded with cookie-cutter stories and dull plots that anything different is a welcome change.

What does this mean for you? Most importantly, it levels out the playing field. That's not to say that you'll be making *Episode I* by yourself in your bedroom, but there are tools out there to get you going. Barriers to entry are lower than ever before. With the Net, anyone can be a viewer, but more importantly, anyone can be a broadcaster.

Look at what IBM did recently. For under $9,000, you can get a full-featured Pentium III–based Intellistation with an Avid Xpress DV editing system, but that's not all. IBM offers the system with a unique feature—free web hosting of your video files on their servers. Getting your content on fast servers is critical to the Net viewing experience, and IBM is offering would-be filmmakers a way to garner increased visibility for their projects.

Just because you wrap a story (or lack thereof) in $100 million worth of 3D effects doesn't mean it will be a success. Conversely, no good story can save bad production values. Do you think *South Park* would be as funny if it were animated more realistically? I doubt it. Part of *South Park*'s charm is in its low-budget look, but the story and characters keep viewers coming back for more. On the other side are films like the *Star Trek* franchise, where messing with the photorealism and believability of the show's imagined future is tantamount to sacrilege. Know your subject matter, your audience, and your content.

As bandwidth increases and people turn more to the Net for entertainment, I expect we'll see the same thing happening to the film and broadcast industry that's happening in music today. Thanks to the Net and technology such as MP3, the music industry is in upheaval. When bandwidth allows, compare film and video to music, and you'll see that the writing is truly on the wall for the coming democratization of the entertainment industry as a whole.

I am fully convinced that the next George Lucas isn't sitting in a boardroom in Hollywood right now. This future star director is at home working on the Next Big Thing as we speak. Yet, as we approach an unprecedented age of personal creative empowerment, there is one maxim that remains eternally true: You can polish a piece of coal all you want, but it still won't become a diamond, no matter how much pressure it (or you) is under.

Chris Tome is technical editor for *3D* and thinks the only thing more depressing than a bad idea with Hollywood behind it is a good idea that never gets done. Sell him on your ideas at ctome@mfi.com.

Postal and Rates *3D* (ISSN 1526-0720) is published monthly by Miller Freeman Inc., a member of the United Newspapers Group, 600 Harrison St., San Francisco, CA 94107, (415) 905-2200. Domestic subscription rate for one year is $29.95. Canada/Mexico $39.95. All other foreign $69.95 (via airmail). Back issue price is $9.00 (domestic and foreign). Periodicals postage is paid at San Francisco, CA and additional mailing offices. **Postmaster:** Send address changes to: *3D*, Circulation Dept., PO Box 5031, Brentwood, TN 37204-5301.

NEW FROM PLAY!

VERSION 2.9 includes new breakthrough modeler

Enterprise Image rendered with Electric Image
"Enterprise" imagery courtesy of Paramount Pictures, Inc. Image TM ® and © 1995 Paramount Pictures. Used in the filming of the movie Star Trek: Generations.

ElectricImage

The World's Fastest 3D Modeling, Rendering and Animation System

From Hollywood's leading motion picture studios to video post facilities everywhere, Time is Money. That's why the world's top broadcast designers, visual effects artists and producers are turning to Electric Image. From last year's Star Wars re-releases to Star Trek: First Contact; from Men In Black to Titanic, Electric Image is the one tool that professionals rely on. Besides its phenomenal impact on Hollywood's top feature films, EI is the 3D system of choice for broadcast graphics, corporate identity and commercial projects. In fact, Electric Image is seen on-the-air somewhere in the world, every minute of every day.

Now Play is proud to unleash Electric Image 2.9. Based on over ten years of trial-by-fire production at top facilities, this breakthrough software combines features of the previous film & broadcast versions and brings many new advanced state-of-the-art features to the desktop for the first time. EI 2.9 is taking the industry by storm with its record-breaking speed and flawless image quality. Whether you're directing the next Oscar-winning blockbuster or creating news graphics on an hourly deadline, Electric Image is the only technology that gives you the creative freedom, uncompromised image quality and breathtaking speed you demand.

Available now for Power Macintosh and soon for Windows NT, Sun Solaris, and SGI Unix.

Complete Film Version Now
$2295.00
was $7,500

MODELING

Hybrid Solid / Surface Modeling
ÜberNURBS
Traditional Surface Methods
Advanced Mesh Tools
Boolean Operation
Bevel Tools
Splines

RENDERING

Anti-Aliasing up to 64k by 64k
Procedural Texture Maps
Clip Maps
Displacement Maps
Texture Animation
3D Camera Projection Maps
Volume Light Rays
Depth Buffer Shadows
Fog and Smoke
Depth of Field
Flexible Glows
Foreground Image Compositing
Stereoscopic Capability

ANIMATION

Function Curve Editor
Bones
Inverse Kinematics
Motion Capture Data
Object Morphing
Network Rendering
Smart Deformations
Motion Blur
Mr. Nitro Particle System
Third Party Plug-Ins

PLAY®
NEXT GENERATION COMPANY

Call **1-714-556-0333**
Visit **www.play.com**
For more information on the Electric Image program visit www.electricimage.com

Play South
3 Imperial Promenade Suite 400
Santa Ana CA 92707